12-18-2020 10:45 AM

Protecting Health Data in a Pandemic: A Systematic Adversarial Threat Analysis of Contact Tracing Apps

Leah Krehling, *The University of Western Ontario*

Supervisor: Essex, Aleksander, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

Abstract

In this thesis centralized, decentralized, Bluetooth, and GPS based applications of digital contact tracing were reviewed and assessed. Using privacy principles created by a contingent of security and privacy experts from across Canada, a metric of assessing an application's privacy was created. An attack tree was built to assess the security of the contact tracing applications. Eighteen attacks were theorized against contact tracing applications currently in use. An application's vulnerability to the attacks was measured using a scoring system developed for this purpose. The results of the security scores were used to create a metric for assessing the security of contact tracing systems.

Five contact tracing applications were assessed using developed privacy and security metrics. The results of this assessment are that for privacy and security a centralized Bluetooth model with added location functionality scored low. While in privacy a decentralized Bluetooth model scored high. In security, the centralized GPS model scored high, while having only a fair level of privacy.

Keywords: De-identification, Health Data, Contact Tracing, COVID-19, Privacy, Security

Summary for Lay Audience

The digital world is growing larger every day. Everything we do that involves a computer or the internet generates data points. These data points are collected and stored. Together all of this data forms a picture of your life. With it models that can predict your behaviour can be created. There is a lot of power sitting and waiting to be used.

The power of this data can be used for good. To create models that can diagnose illness or determine the best treatment plan for an individual. It could also be used to harm people. The same medical record that together with others could create a treatment breakthrough for a mental disorder could be used to discriminate against someone with that disorder, losing them their job. Health data is private for good reasons.

Determining the best practices of keeping health data private provides those tasked to do so with the tools they need. The first section of this thesis is a review of the state of health data privacy. This leads to an overview of the best practices of the field.

Then the specific health data problem of contact tracing is tackled. Both the privacy and security of contact tracing applications is important. The privacy of the application was measured using privacy principles created by a contingent of security and privacy experts from across Canada. From these privacy principles, a metric for assessing an application's privacy was created.

To assess the security of the contact tracing applications eighteen attacks were theorized. These attacks were then applied to the systems that the application's use. An application's vulnerability to the attacks was measured using a scoring system developed for this purpose. The results of the security scores were used to create a metric for assessing the security of contact tracing systems.

Five contact tracing applications were assessed using developed privacy and security metrics. The results of this assessment are that for privacy and security one out of the five was ranked as low, one was ranked as high, and three were ranked as medium. This means that of the five applications scrutinized four out of five have privacy or security concerns. As these applications are intended to be used across the globe by everyone for the safety of the populace these concerns are important to address.

Acknowledgements

I would like to thank the efforts of my advisor Dr. Aleksander Essex. He is an excellent mentor and it is only through his guidance that I was able to learn so much in the last two years. I am incredibly grateful for the support he has provided. Thank you for every insight, pep talk, and for responding to the email of a fourth year undergraduate student in need of an advisor.

I also would like to thank Elena Novas and Zeev Glauberzon. Their support, insight, and efforts are greatly appreciated. Thank you for answering all of the many questions I had about data privacy and your unfailing patience while teaching me.

To the rest of my friends and family thank you for your support, your conversation and your impeccable ability to keep me laughing.

Finally I would like to acknowledge my dog, Naga for all of the work she did as emotional support. Without her this may have been completed but it would have taken a lot longer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The total amount of digital data in the world is estimated to be 44 zettabytes in 2020 [48]. In 2013, there were 4.4 zettabytes of data. For reference the entire Encyclopedia Britannica in its final volume was one gigabyte of disk space [92]. One zettabyte is a trillion gigabytes. Data has been important to humans since the creation of language, the library of Alexandria in Ancient Egypt was the largest store of data in 300 BC [48]. Humans it seems, have always believed that the collection and analysis of data was the key to something. That data was the foothold from which civilization could rise above its current problems and become greater. Knowledge is, after all, power.

The Information Age arriving in the 1970's changed humanity's understanding of data. With the creation of transistor technology data has become digital. It was the development of the Internet and the Internet's growth over the last two decades that truly and completely changed how humans thought about data. Prior to then, people had to actively create data. Data had to be actively collected from research, written into books, copied, and read through to be utilized. Now data is generated by the systems we are using every day. Data is created and stored without humans ever having to touch it.

Humans still use data to create solutions to problems. When the amount of data available became too large for people to analyze they turned to the machines that had generated so much of it for the solution. Machine learning was created. Computers could be used to analyze the hard drives full of data. Using new analytical methods data could be utilized to guide solutions and train predictive models. From the analysis of data, knowledge could be created, however, power can be misused.

Some information, even in the digital age, needs to be kept private. Military bases for instance sometimes need to keep their location and movements of soldiers secret. The location data from soldier's Fitbit watches could be used to learn such things [145]. Data that needs to be kept private is not restricted to state secrets. Personal data can be powerful as well. In 2017 a data breach of Equifax (one of the largest credit bureaus in the United States) exposed the personal data of 147.9 million people [29]. The data Equifax held could be used to steal identities, steal money, and in the long run hurt every one of those people.

The medical industry has been no different from any other in recent decades. Data has been used to create solutions and solve problems. Artificial intelligence is being used to create predictive models that will advance health care in the future. Faster diagnosis, a better understanding of illnesses, a better understanding of the efficacy of different treatments, these and more are all possible results of leveraging the health data that is available [65]. The healthcare industry collects a huge amount of healthcare data from patients through clinic appointments, hospital visits, and medical studies. More than just hospitals and universities are collecting this data as well, IBM [65], Apple [66], Google [60], Microsoft [85], and Amazon [24] all have some form of healthcare research or service or application.

All of the data generated by every one of the 7.5 billion people on this planet needs to be protected because every single one of those people deserves to be protected. Information is knowledge and knowledge is power. Privacy is about how much power we give to those we trust. Security is about keeping that power in the hands of only those we trust. The only way to ensure that our data is private and secure is to test and scrutinize its handling.

1.1 Motivations

On January 30^{th} 2020 the World Health Organization's (WHO) Director-General declared the novel coronavirus outbreak a public health emergency of international concern, the WHO's highest level of alarm. On March 11^{th} due to the concerning levels of spread and severity of symptoms the WHO assessed that the coronavirus disease of 2019 (COVID-19) could be characterized as a pandemic [131]. The Director-General said that "all countries can still change the course of this pandemic" if they "detect, test, treat, isolate, trace, and mobilize their people in the response" [55].

The pandemic of 2020 is the first outbreak to spread across the globe since the dawn of the digital age. Naturally, the digital world has turned to digital solutions. One of which is to take

the detect and trace parts of the pandemic response and automate them. Digital contact tracing is a new idea for the health care field. When new tools are created whose purpose is to collect data and trace people the privacy and security field need to move quickly. To pump the breaks when the momentum is picking up. Not to stop the advancement, but to move with the caution the icy road of data collection requires to navigate safely.

The goal of this thesis is to create a metric to use against digital contact tracing applications. This begins with a review of the entire field of health care data. Once the best practice of how to treat the different data types that are a part of a modern health record is established the field of digital contact tracing can be assessed. Contact tracing applications have the potential to assist the pandemic response and impact the spread of the virus. However, any application with the intention of tracking its users needs to be thoroughly vetted before implementation. This thesis intends to create that vetting process.

1.2 Contributions

The intention of this body of work is to layout first the current landscape of health data protection. A complete review of health data privacy was performed to determine how data is protected. This review was intended to determine not only what methods are being used but what methods have been shown to be broken and what overall the best practice of health data privacy should look like. This forms part one of this document.

Part two takes this knowledge and applies it to a new aspect of the health care field. Digital contact tracing. Using principles of data privacy fifty-five of the contact tracing applications being released by governments around the world were assessed. This information was then used to create a metric for measuring the level of privacy their users have. Then a security assessment of the apps was performed. Through analysis of the applications' operations, eighteen different potential vulnerabilities were used to test security. From this security assessment, a metric for measuring the level of security of the apps was created.

Governments and citizens could use the assessment tools created in this thesis to scrutinize contact tracing applications prior to their adoption. Thus, proactively protecting their citizens or themselves from the dangers inherent to unsecured health data. The developers could use these metrics to determine how they should design the systems or improve them. Making the systems more secure moving forwards.

1.3 Organization of Thesis

The rest of the thesis is organized as follows.

- **Part 1: Systematic Review of Health Data Privacy**. The chapters in this section form a detailed review of the state of health data privacy.
  - **Chapter 2: Background of Privacy and Health Data**. This chapter details, how they can be privatized, how they can be attacked, and the dangers of those attacks succeeding.
  - **Chapter 3: Review of Health Data De-identification Attacks**. A review of attacks performed to determine the identity of the individuals from released data sets that had privacy claims was performed. This chapter details that review.
  - **Chapter 4: Observations from Review of De-identification Attacks**. The review of attacks is applied to future data with a set of guidelines on what to do when removing identities from data sets.
- **Part 2: Security and Privacy of Patient Contact Tracing**. The chapters in this section form a detailed review of the privacy and security of contact tracing apps. A method to determine the risk to the populace when digital contact tracing is implemented with the intention of being used country-wide is developed.
  - **Chapter 5: Background on Contact Tracing**. This chapter details background information on contact tracing. Detailing what it is and what it is becoming.
  - **Chapter 6: Contact Tracing Schemes**. A detailed overview of the operation of the most popular digital contact tracing schemes.
  - **Chapter 7: Methodology of Developing a Tool to Assess Contact Tracing Applications**. In this chapter, the methodology followed to create an assessment tool for contact tracing apps is detailed.
  - **Chapter 8: Assessing Contact Tracing Applications**. This chapter contains a complete privacy analysis of the contact tracing apps released in over fifty countries. Comparing them against the determined standards and ranking them accordingly. Then a security review of five of these apps is performed. The assessment system of both the privacy and security of the apps is developed and tested.

– **Chapter 9: Discussion, Future Work, Conclusions**. A concluding chapter that discusses the contribution of this thesis and the scope of future work from it as well as some concluding remarks.

Part I

Systematic Review of Health Data Privacy

Chapter 2

Background of Privacy and Health Data

Data has become a commodity of the modern world. Both companies and research groups want to use the vast amounts of the data that has been collected from people to create innovative solutions, perform groundbreaking research, or optimise their designs. A common use of this data is to train machine learning algorithms with it to make them better predictors of customer behaviour. For example, companies like Amazon, Apple, Google, and Microsoft [33, 143, 144] have all fed voice data to their software-based assistants (Echo, Siri, Google Assistant, and Cortana respectively) to teach them to better understand human speech patterns and accents. Voice data is not the only type of data that can be leveraged and there are many different data types and structures that can be used. Some examples being geolocation data, cell phone records, social media networks, medical data, and browsing data. The data collected is diverse in type and form, making even single data sets complex, as they can contain any combination of this variety.

Much of the data that has been collected has the potential to reveal the identity of the people it is about. The information would initially contain personal details, like data about online shopping habits that enter a database connected to an account and thus contains the name, email address, phone number, credit card information, etc. of the customer. Due to this, to protect people's privacy, many countries have legislation in place to limit the collection and control the distribution of data about its citizens. For example, Canada has the federal Personal Information Protection and Electronic Documents Act (PIPEDA) [164], which places protections on personally identifiable information collected about Canadians.

De-identification refers to the processes used to separate someone's identity from the data collected from them to prevent their identity from being revealed through observation and

analysis of the data, or linking data sources together. The idea of data de-identification then is to reduce the risk of connecting the data to the originating individual to a statistically insignificant amount. Under legislation such as that in Canada, data that has been de-identified is no longer considered protected personal information because of the idea of there being a statistically low risk of exposure [164].

There is a growing view that the risk assurances of de-identification are flawed, and that de-identification does not work [111]. Several notable cases have shown that supposedly de-identified data could still be used to re-identify individuals [89]. Cases of this occurring cast doubt on whether de-identification can protect an individual's privacy. Testing the accuracy of this view is a goal of chapters 3 and 4 of this thesis.

2.1 De-identification Methods

The field of research into different methods of lowering the re-identification risk is quite broad. Not only are there are many different forms of data but each data set is different in its collected attributes. For example, geolocation data sets, one from a vehicle's GPS and another from a social network site. The data set generated from the vehicle might contain only two pieces of information, the time and position coordinates. The social network might contain the same timestamp and coordinate information, but also the user's name, age, the content of their post on the site, and other information outside of the scope of what one would consider geolocation data. This extra information can help inform researchers that intend to use the data to learn something about the populace. For example, social media data could be useful to the planning of public transport routes by providing the movement patterns of specific age groups. Due to this variance between the data within sets of the same "type" often data sets need to be considered individually. The unique combination of features in a data set could reveal more information than the features of another data set would.

De-identification literature typically splits data features into two types: direct identifiers and indirect identifiers. Direct identifiers include anything that on its own can be tied to an individual's identity, a name, social insurance number, account number, etc. These are values that are unique to an individual and can act as their identity in some contexts. An indirect identifier is a value that on its own is not enough to identify a person; date of birth, gender, race, etc. However, when used in combination they create indirect identifier sets that can be used to identify a person. These can also be called quasi-identifiers. A third category could

be non-identifiers, values that cannot identify a person, though it can be difficult to draw the line between them and indirect identifiers, as generally the larger the indirect identifier set the easier it becomes to identify an individual, even when the information seems inconsequential. Thus, except for cases where the attribute is not externally available for an adversary to exploit, the line between non-identifiers and indirect identifiers is difficult to set.

Due to the variance in data types and data sets, different actions need to be taken to de-identify data sets. These actions can be generally grouped into six types of manipulations: suppression, masking, generalization, perturbation, aggregation, and access control and monitoring.

Suppression.

Suppression is the removal of some aspect of the data set entirely. This could include completely removing a field or column in the data, such as the individual's name, or removing specific entries in the data set. Outliers in the data set that are too extreme might be suppressed for a variety of reasons, one reason being that the uniqueness of any outlier makes that individual more likely to be identified in the data set. Providing the same privacy assurance to the outlier as all other individuals in the set could require manipulations that reduce the research utility of the data set too far, as the section on generalization explains. Thus removing the outlier can be the more prudent choice.

Generalization.

Generalization is a decrease in the granularity of the information, resulting in the information being less specific. This could be changing the time stamp from the second that it occurred to the minute, a five-minute interval, or more as required. It could also be changing the location from the exact address to the city block or general neighbourhood. The chosen generalization level for a feature in the data set is applied to that feature for every entry in the set. The risk with generalization is that increases in generalization are also decreases in utility. The greatest amount of privacy that could be provided to individuals would be to make all of the entries the same, or not include any at all, which would not provide much information for researchers to use. For example, if every location is generalized to the country but the researchers need to know which provinces or even townships an illness is most prevalent in the data has been too generalized to be useful to them.

Masking.

Masking is when the data is left in the set but is obscured so that the original values cannot be readily obtained. This is usually used on direct identifiers. Sometimes this is done to make it easier for researchers to identify entries that have been made by the same individual, creating a perpetual identifier used through the data set that allows them to trace a single person. Depending on the intended usage, the data may need to be able to be linked back to the original individuals at a later time and so a value connecting the entry back to the data set containing their identity is required. Pseudo-anonymous identifiers do not appear to be an identifier, but information can be linked to them by the service providers or site administrators. For example, in cell phone records for many users, every individual would require a perpetual identifier so that the logs for the same person can be grouped together. This can be done using techniques like encryption or, in limited cases, hashing on the original value.

Perturbation.

Perturbation is to alter the reliable accuracy of the value, typically seen as adding noise to the system. This is not a generalization, as the granularity does not need to change, but instead, the value itself is slightly altered. An example is shifting GPS coordinates randomly by a small amount so that the exact location cannot be assumed to be reliable. The goal is not to alter the result of any analysis but the reliability of a single value so that it cannot be used to concretely learn someone's identity.

Aggregation.

Aggregation is the process where raw data is collected or grouped together. In some cases only statistics about the data may be released, in others, attributes or entries may be combined. This way instead of revealing the entire data set collections of statistics derived from the data set or information about small groups within the data set can be revealed. For example, if analysis of the raw data shows that 50% of men over a certain age living in a specific township have a disease that is being studied, only that statistic would be released. For geolocation data, this could also mean showing the popularity of certain locations rather than the actual mobility traces of the individuals, or grouping multiple traces together into average movement patterns.

Access Control and Monitoring.

Access control and monitoring is the limitation of access to the data set. For this thesis, this will be considered anything that actively limits how someone can access the data, or shields the data in some way, as opposed to simply a signed agreement to not misuse the data upon receiving it. Things like the data only being query-able instead of allowing access to the raw data, or limitations on who has access to the data, or parts of the data. The data set might also have monitoring on it that records who is accessing it when, for how long, a record of queries made to the data, etc.

2.2 The Threat of Re-Identification

The purpose of de-identification is to strip the personal identity of the data source from the data itself. This is done so the data can be passed to researchers or made publicly available with minimal risk to individual privacy. In encryption literature, there is the concept of encryption being broken. Essentially if currently available computers could guess the encryption key within a relatively short amount of time, then the encryption is not secure. However, if it would require more computing power than is currently conceivably possible the encryption can be considered secure. A similar concept is used in de-identification research. "Factual Anonymity", sometimes referred to as "Practical Anonymity," says that if it would require an excessive amount of time, expertise, manpower, and expense to re-identify individuals of the data set, then the set can be considered "factually anonymous" [79]. From factual anonymity, it becomes clear that the study of re-identification attacks should take into account the expertise required of the attacker, as well as the information that they have available to them or the cost of the type of data that might be required to link to someone's identity.

There are some commonalities between re-identification attacks. Typically they require information outside of the de-identified data set to match with individuals within, an individual that then appears in both data sets, common information about the individual in both data sets, and enough information to be statistically sure that a match is correct [6].

2.2.1 Identity Disclosure

Identity disclosure is the full re-identification of one or more individuals within the data set. The attacker has somehow re-identified these individuals despite the de-identification efforts

that were made to prevent this. There are many ways this can occur, and the accuracy of the re-identification can vary depending on the attack method and the original data set. Though it should be noted that the miss-identification of an individual as someone in the data set can also be harmful to them as many of the negative impacts of identification could still occur to them in real life despite the inaccuracy of the attack.

2.2.2 Attribute Disclosure

This can also be referred to as "Homogeneity Attacks" or "Inference Attacks" when discussing *k*-anonymity, which is described in Section 2.3.1. The idea of this type of privacy loss is that the attacker learns something about the individual that is not public knowledge without fully re-identifying their entry within the data set. For example, if an attacker is trying to discover an individual's illness and has access to health data they know to contain that information, then they are likely able to narrow down the possibilities of who the individual is. Say they are looking for a male age 40, removing anyone who does not meet this requirement might leave them with 10 possible individuals. If all 10 individuals have the same illness, then the attacker has learned what they wanted to without ever re-identifying the exact entry of the person they sought information on.

2.2.3 Internal Attacks

These are attacks that originate from within the data set. The attacker starts with the data set and attempts to find unique individuals within the set. The methods will vary depending on the type of data. In geolocation data, an attack will look for unique movements or outliers. Once individuals are isolated, the attacker will take the information available and compare it to public information to identify someone. With geolocation data this could mean analyzing the mobility traces to find a likely home address that can be searched in public databases, thus providing them the identity of these individuals.

2.2.4 External Attacks

These are attacks that originate from outside of the data set. An attacker might be someone who knows that a particular individual is within the data set and seeks them within it. It could be someone who knows the individual personally or, in the case of public figures, simply has background information on the individual. In this case, the attacker would access the database

and use the background information to isolate the individual they are seeking. For geolocation data, if an attacker knows where an individual was at a specific time, they could use that to isolate their mobility trace and learn their movements over the entire time coverage of the data set.

2.3 De-Identification Measures

There are two main measurements of privacy in the literature that can be used to determine how much privacy is being provided for individuals in the set. These are k-anonymity and differential privacy. k-anonymity looks at the uniqueness of every individual in the data set, while differential privacy balances the amount of accurate information revealed with the error created in the analysis of the data.

2.3.1 k-anonymity

k-anonymity is a de-identification measure that was developed by Sweeney [151]. The general idea is that every individual should be indistinguishable from $k-1$ other individuals in the data set when $k > 1$. To make individuals indistinguishable, information is typically either suppressed or generalized. Any generalization to an attribute must be uniform throughout the data set. It also might require the removal of extreme outliers from the data set as they may be so unique that generalizing the entire data set would remove too much information. By doing this it becomes less likely that an attacker could isolate a single individual in the data set.

There are also multiple extensions of k-anonymity; ℓ-diversity [95], p-sensitive k-anonymity [163], and t-closeness [86]. These extend the requirements of k-anonymity based on examinations of when the asserted k-anonymity promise of privacy might break. The methods of these papers should be considered when looking at what constitutes best practice.

An issue with k-anonymity is that since alterations to the data set must be made computational resources are required to perform these operations. Though much of the manipulation could be automated, often manual manipulations or checks on the anonymity level still need to be made, and the alterations could make the data significantly less useful. By generalizing the data the caretaker could make the data too generic to be useful for the purpose of the release and would then have no reason to release the data.

2.3.2 Differential Privacy

Differential privacy was introduced by Dwork et al. [37]. The basic idea is to balance the level of privacy provided to individuals in the data set with the accuracy of the data. Privacy is created by introducing unreliability to the singular attribute entries by adding noise to the values of the attributes. If the noise is calibrated carefully then the results of the analysis will be within an acceptable error of the unchanged data so the result of the intended analysis of the data is unharmed. However, anyone trying to identify someone will be prevented from doing so because there will be no way to tell if the values are correct so long as the noise that is introduced is not predictable.

The exact method of introducing noise is up to the data custodian to determine. As well as the level of privacy and error. Differential privacy is often discussed when dealing with query systems, as the noise can be added to the data at the time of the query and adjusted according to the specific query.

An issue with differential privacy is that, similar to k-anonymity, it is difficult to say what value the privacy/accuracy level should be set to. If too much noise is added to the data then the analysis might be unreliable, and the amount of noise that is added depends on the values themselves. If there is little variance in a value across all the data entries, then only small amounts of noise can be added. It requires knowledge of the data, and an understanding of the attributes themselves to tune the noise. This also means that each feature of the data set may require individual tuning. In a data set that has many features this can be computationally and time-intensive. As well, even with the balanced noise, the more queries that are performed the more data that is released, and potentially enough data is released to re-identify someone if the queries are not limited.

2.4 Tools of Re-Identification

Though this list is not exhaustive it provides a basic overview of the common methods and tools that attackers will use to identify people within a data set.

2.4.1 Linkage Attacks

These are attacks that are performed by connecting the information from two data sets. It is a very common attack, particularly when public records of names and addresses are available

to the attacker. This could also involve linking two de-identified data sets that originated from the same raw data (or not) to gain enough information to identify the individuals that appear in both data sets [150]. These attacks can be done manually or be automated in various manners.

2.4.2 Machine Learning Models

Common with internal attacks, machine learning models can be trained to analyze the data and determine the information that can be used to identify individuals. For example, Geolocation data can be analysed with clustering models to find an individual's most likely home address or workplace [36]. It could also be taught an individual's patterns and then used to find that individual again within other data sets that are released [46].

2.4.3 Markov Chains

A Markov Chain is a statistical model. Specifically, it is a model that describes the probability of the next state of the system based on its current state. Markov chains are often used with geolocation data to create statistical models of individuals' movements. It could be used to make predictions of where an individual will be and when for an attacker to identify them in person [155].

2.4.4 Reversing Masking

This depends on the type of algorithm used to mask values within the data set. In some cases, it is possible that the actions performed on the original data can be reversed and so the original value can be discovered by the attacker [31]. If masking is done by performing reversible mathematical operations on the raw data the attacker could discover the steps of alteration and reverse them [84]. Or if hashing was performed it might be possible for an attacker to brute-force match the hash with its originating value [133]. Even encryption could be vulnerable depending on the type and its implementation.

Chapter 3

Review of Health Data De-identification Attacks

3.1 De-Identification Review Methodology

A systematic review was performed of the relevant evidence demonstrating successful re-identification attacks on data sets that had some transformations applied to hide the individuals' identity. Papers and articles from a wide array of communities reporting on such attacks were examined, including statistics, computer science, and health informatics. As well as journalistic explorations.

3.1.1 Search Method

Articles were searched using the general terms "anonymization", "de-identification", and "re-identification". Broad search terms were chosen to ensure that we did not miss any relevant publications. The searches were performed on PubMed, IEEE Xplore (the online library of the Institute of Electrical and Electronics Engineers), the ACM Digital Library (the online library of the Association for Computing Machinery), ScienceDirect, Research Gate, Springer, and JSTOR online repositories, and the records for all relevant English language articles were obtained for further consideration. The resulting set of articles was augmented with previously known articles, identified through targeted searches on Google Scholar (e.g. for specific authors), and articles identified through the reference lists of the included research. Technical reports, presentations, and news articles with re-identification events performed by the journalist were also included.

3.1.2 Inclusion/Exclusion Criteria

Many articles were identified through the search performed. The article titles, keywords, and abstracts were screened and the primary inclusion criteria was that the researchers had taken a real data set that was available to them and worked to identify the individuals within the data set or break the privacy promise made by the releasing party. Emphasis was placed on the data set being real and was not placed on the releasing party being the one to perform the de-identification. In some cases, the data set was created by the researchers themselves, de-identified using the same standard as previous releases of similar data, and then attempts to re-identify the individuals were made. Since this was done to obtain consent from the individuals whose data makes up the set this did not exclude a paper from the scope of this review.

3.1.3 Data Abstraction.

The following criteria were used to summarize each eligible study: (a) the type of data included in the attacked data set, (b) the method of de-identification used, (c) the profession of the adversary, (d) country of re-identification, (e) the proportion or number of individuals re-identified, and (f) whether the re-identification was verified. The first four criteria were used to characterize the nature and scope of successful re-identification attacks, whereas the last two are quality indicators for the attack. While not comprehensive, these criteria were believed to provide a descriptive summary by which the nature of the attacks could be examined and compared.

Type of Data Included.

Not all data sets are structurally the same, and each type of data set requires its own de-identification and re-identification methods. The de-identification of health data requires different actions than geolocation data, along with browsing data. To fully explore the space and issues with the success of de-identification the type of data used must be considered.

The Method of De-identification Used.

Not all de-identification methods are the same, different methods and measures are used for different types of data and there are few official standards in place. Health data has the most standards about what constitutes a de-identified data set in particular The US Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. Which provides a description of how to de-identify health data to meet the privacy requirements of US law. This standard has

been applied in other jurisdictions, research organizations in Canada will cite it as best practice [42] and perform de-identification according to its provisions as they often have overlaps with US agencies [40].

Outside of these set standards, a standard of best practice is often used which mimics many of the edicts of HIPAA. The removal of names, complete dates, and exact addresses. Though HIPAA extends these further with requirements of 'experts' to perform the de-identification, that the risk of re-identification is 'very low', and that the methods used are documented.

Due to the open interpretation in some areas of the standards that do exist and the differences in the methods used to de-identify different types of data the type of de-identification used was examined. Whether or not this data was 'defensibly' de-identified will be discussed and determined with the standards more rigorously considered.

The Profession of the Adversary

Who is re-identifying data sets helps to determine how widespread re-identification attacks are. If people of many different professions, skills, and resources are launching successful re-identification attacks this may indicate how easy these attacks are to perform or learn to perform.

The Country of Re-identification.

This refers to both the country of the adversary and the country where the individuals of the data set reside. This is considered because some countries make population databases readily available for free or for a modest fee. A good example of such publicly available population databases are state-level voter registration databases in the US [10]. There is also a thriving industry specializing in the creation and sale of databases containing personal information about the population, making a successful re-identification attack on a de-identified data set more likely [134].

The Percentage/Number of Individuals Re-identified From the Data Set.

The percentage of individuals from the data set that can be re-identified is an indication of the success and severity of the re-identification attack. A large percentage of records in a database being re-identified is a more severe attack than a single individual being re-identified. For

this review, the success rate was based on what was reported within the documentation of the attack.

Re-identification has been verified.

Once the adversary has linked various records from the de-identified data set to real individuals there is the possibility that some of the results are false positives. Re-identification is probabilistic. Often it is based on the probability that there is only one person with the specific characteristics that are being analysed. Even if the probability is high, without some form of verification there is the potential for false positives. Data sets also have quality problems. There is no guarantee that the information about an individual in the data set is completely accurate, particularly for values that change on a semi-regular basis. The date of birth might be entered incorrectly, or the phone number could change, people can move addresses. There is the same issue of reliability with the reference data that the attack may be used to perform the re-identification.

The adversary could verify that the re-identification is correct using some additional information. This is simple to do when the adversary has created the original unchanged data set or otherwise has access to it, or if the data was synthetically generated. In some cases verification could require contacting the individuals directly if the original data set is not available, these are direct methods of verification.

In some situations, verification could be indirect. If the individual is a known public figure then there may be information publicly available about them from other sources that provide the required verification.

Though it is not possible to know with certainty if the chosen record is the correct one without verification, in some cases, this is not possible. It could be that directly contacting the individuals would be too great a breach of privacy, depending on the type of data that has been de-identified. If the data custodian is not able or unwilling to grant access to the complete data set or confirm internally any identifications that have been made. In such cases, a thorough examination of the error rate of the re-identification method should be done by researchers.

For this review, if verification of some kind was performed it is indicated in the summary table. If instead error was calculated this is indicated as well. If there was no mention of verification, a false positive rate, or error calculation then it was assumed that these were not performed.

3.2 Results From the De-identification Attack Review

There were 60 studies included in the final de-identification review of this thesis. There were some that were excluded because though the study the researchers were performing was similar it did not go so far as actually finding the identities of individuals. These include papers in which likely homes were found [4] [146] [70]. Some studies linked anonymous users visiting a site from different browsers so that they could be identified [23] [13]. Or studies that performed inference attacks to learn traits of users from data available [81] [20].

3.2.1 Notable Observations

There were several notable observations made during the review that should be highlighted:

The provided detail of attack varied.

Some of the reports had in-depth and detailed descriptions of exactly how they attacked the data and performed their re-identification. While others provided little information on the specifics of the failure in de-identification. In the case of [31] this was because of the nature of the data, as it was medical data with a faulty implementation of an encryption algorithm for masking. Even fewer directly indicated the secondary data used to identify individuals, often referencing public voter lists. This makes it difficult for researchers to verify the results received by those who performed the original attack.

Majority of the attacks were performed by researchers.

Only three of the attacks found were performed by groups outside of a University or established research centers like Microsoft Research, or IBM Watson. In one case [133] the individual is an engineer who also reports on privacy and software topics. In the other cases [39] [8], journalists performed the attack. In the case of [39], a data scientist was brought in. This was also the only attack that indicated the information was not received through public release, or in agreement with the data custodian to test the de-identification.

Majority of the Attacks were in the United States.

Of the attacks that were studied most of them were performed by US researchers and the data was about US Citizens. This likely reflects a greater availability of public and semi-public

information available to researchers for performing re-identification attacks. The success of many of the attacks was noted to be jurisdiction dependant because of the differences state to state in the availability of public information. It could also reflect a larger group of researchers in the field for this region.

Most of the attacked data sets were not de-identified to existing standards.

There were only three attacks that mentioned the data having been de-identified to HIPAA standard [149],[71], and [11]. Though there is the potential that some of the health data sets were de-identified to HIPAA standard and it was not specified. Three mentioned that an implementation of differential privacy had been used as a measure for the amount of data that was being released [73], [156], [87]. There was only one that mentioned the data following a k-anonymity framework [148]. It should be noted that except for [11] all of these papers used data sets from the last 4 years.

Often the issue seems to be that direct identifiers were removed, however, what exactly is considered a direct identifier varied. As in the case of [91] which left the 3 digit ICD-9 diagnosis codes, which in many cases where specific enough to be unique when compared to resources that mentioned the health issue. But best exemplified by [103] which left phone numbers in the call records that could simply be looked up. This issue repeated itself many times with collections of quasi-identifiers that could be compared to voter lists and identify individuals.

This is also apparent with the geolocation data. For the majority of the attacks, identifiers had been removed or masked. However the mobility traces were unique enough that individuals could be isolated and their home address, or home-work pairs that could be matched to public data.

A similar problem existed with the attacks on browsing data. Both of them [39] and [8] contained only time stamps and searches made, however, the URLs themselves contained addresses and other personal information that made identifying individuals possible.

3.2.2 Health Data

Attacks on medical data included personal information, illness history, prescription history, procedure history, attending physician, and genomic data. For a majority of health data exam-

Table 3.1: Summary of Re-identification attacks

	Study	Year	Adversary	Country	Suppressed	Masked	Generalized	Perturbed	Aggregated	Controlled	Failure	Success Rate	Verified
Health Data	[71]	2018	Researchers	USA	●	●	●	○	○	○	Theory	N/A	No
	[159]	2018	Researchers	TR,USA	●	○	○	●	○	●	Theory	90%	No
	[110]	2018	Researchers	USA,CHN	●	○	○	○	○	○	Theory	94.9%	Yes
	[30]	2016	Researchers	AUS	●	●	○	●	○	○	Theory	80%	No
	[31]	2016	Researchers	AUS	●	●	○	●	○	○	Implementation	100%	Yes
	[43]	2013	Researchers	CAN	●	●	○	○	○	○	Theory	30%	No
	[154]	2013	Researchers	USA	●	●	○	○	○	○	Theory	97%	No
	[153]	2013	Researchers	USA	●	○	○	○	○	○	Theory	43%	Yes
	[83]	2012	Researchers	USA	○	●	○	○	○	○	Theory	95%	Yes
	[80]	2010	Researchers	NLD	●	○	○	○	○	○	Theory	99.4%	No
	[11]	2009	Researchers	USA	●	●	●	○	○	●	Implementation	60%	Yes
	[44]	2009	Researchers	CAN	●	○	○	○	○	●	Theory	N/A	Yes
	[91]	2009	Researchers	USA	●	○	○	○	○	○	Theory	96%	Yes
	[96]	2006	Researchers	USA	●	●	●	○	○	○	Theory	70%	No
	[98]	2003	Researchers	USA	●	○	○	○	○	○	Theory	100%	Error
	[152]	2003	Researchers	USA	●	○	○	○	○	○	Theory	6.1%	No
	[151]	2002	Researchers	USA	●	○	○	○	○	○	Theory	N/A	Yes
	[97]	2000	Researchers	USA	●	○	○	○	○	○	Theory	98%	No
Geolocation Data	[46]	2019	Researchers	CHN,USA	●	○	○	○	○	○	Theory	50%	Yes
	[54]	2019	Researchers	CHN,CAN	●	○	○	●	○	○	Theory	90%	No
	[102]	2019	Researchers	BEL	●	●	○	○	○	○	Theory	80%	Yes
	[36]	2019	Researchers	GRC,USA	●	●	○	○	○	○	Implementation	92.5%	Yes
	[156]	2018	Researchers	USA	●	●	○	●	○	○	Implementation	N/A	No
	[100]	2017	Researchers	FRA	●	●	●	●	○	○	Implementation	79%	No
	[155]	2017	Researchers	USA	●	○	○	●	○	○	Theory	N/A	No
	[87]	2016	Researchers	USA	●	○	○	●	●	○	Implementation	N/A	No
	[84]	2016	Researchers	LVA	●	●	○	○	○	○	Theory	N/A	Yes
	[35]	2014	Researchers	USA	○	●	○	○	○	○	Theory	91%	Yes
	[133]	2014	Blogger	USA	○	●	○	○	○	○	Implementation	100%	Yes
	[161]	2014	Researchers	USA	○	●	○	○	○	○	Theory	N/A	Yes
	[52]	2013	Researchers	FRA	●	○	○	○	○	○	Theory	45%	Yes
	[94]	2013	Researchers	CHN,USA	●	●	○	●	○	○	Theory	50%	Yes
	[107]	2012	Researchers	USA,CHL,BEL	●	●	●	○	○	○	Theory	95%	No
	[170]	2011	Researchers	USA	●	○	○	○	○	○	Theory	35%	No
	[50]	2011	Researchers	CHN	○	●	○	○	○	○	Theory	65%	No
	[53]	2010	Researchers	FRA	●	●	○	●	●	○	Theory	50%	No
	[75]	2009	Researchers	UK	●	○	○	○	○	○	Theory	N/A	Yes
	[59]	2009	Researchers	USA	●	○	○	○	○	○	Theory	90%	No

Table 3.1: Summary of Re-identification attacks

	Study	Year	Adversary	Country	Suppressed	Masked	Generalized	Perturbed	Aggregated	Controlled	Failure	Success Rate	Verified
	[109]	2008	Researchers	UK,BEL	●	❍	❍	❍	❍	❍	Theory	80%	Yes
	[82]	2007	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	5%	Yes
	[17]	2006	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	79%	No
Simple	[138]	2019	Researchers	UK,BEL	●	❍	❍	❍	❍	❍	Theory	99.98%	Error
	[149]	2017	Researchers	USA	●	●	●	❍	❍	❍	Theory	28%	Yes
	[77]	2011	Researchers	CAN	●	❍	●	❍	❍	❍	Theory	98%	No
	[58]	2006	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	63%	No
	[6]	2001	Researchers	DEU	●	❍	❍	❍	❍	❍	Theory	14%	Yes
	[150]	2000	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	87%	No
Social	[99]	2018	Researchers	CHN	●	❍	❍	❍	❍	❍	Theory	58%	Yes
	[76]	2014	Researchers	USA	●	●	❍	❍	❍	❍	Theory	95%	Error
	[113]	2007	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	30.8%	Yes
Misc	[73]	2019	Researchers	USA	●	❍	❍	●	❍	❍	Implementation	7%	Error
	[32]	2015	Researchers	USA,DK	●	●	●	❍	❍	❍	Theory	90%	Yes
	[147]	2017	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	70%	Yes
	[39]	2016	Journalists	DEU	●	❍	❍	❍	❍	❍	Theory	N/A	Yes
	[8]	2006	Journalists	USA	●	❍	❍	❍	❍	❍	Theory	N/A	Yes
	[112]	2008	Researchers	USA	●	❍	❍	●	❍	❍	Implementation	80%	Yes
	[49]	2006	Researchers	USA	●	●	❍	❍	❍	❍	Theory	N/A	Yes
	[148]	2018	Expert Witness	USA	●	●	●	❍	❍	●	Implementation	66%	Yes
	[103]	2014	Researchers	USA	●	❍	❍	❍	❍	❍	Implementation	86%	Yes
	[116]	2001	Researchers	USA	●	❍	❍	❍	❍	❍	Theory	75%	No

ined, it was not explicitly stated that the HIPAA guidelines were followed, however for data released from the United States after April 14th of 2003 when it became the federal standard.

The two medical data sets that were explicitly de-identified to HIPAA standards [71] had data about motor vehicle accidents. Due to this, they determined the identity based on limited quasi-identifiers using news reports about accidents that contained the individual's identity.

[11] compares HIPAA protections with another "Limited Dataset" standard that removes identifiers and has a contract signed that dictates proper use of the data set. Depending on the state the individuals lived in from the safe harbour protected data they could identify 0.01 to 0.25% . From the limited data set 10% to 60% . The difference between states was because they used the voter information, the detail of which is state-controlled.

Other attacks used newspapers as their secondary source as well [153] used articles with the term "hospitalized", [43] used publicly available obituaries. This one had genomic data [154] and used public record

Two studies looked at health data in Australia. Both examined the same data release and were performed by the same researchers. [31] is reported to have discovered a vulnerability in the encryption used for the patient insurance numbers. Due to the very sensitive nature of this data the government was informed, the data was taken down and the information about the encryption used that had previously been available on the website was also taken down. [30] then took the same data set and attacked it without using the encryption vulnerability.

3.2.3 Geolocation Data

For the geolocation data there were two different general types that were attacked. Some of the geolocation data was continuous and made up of mobility traces [46], [35], [102], [156], [155], [133], [161], [94], [107], [50], [53] from a variety of sources. Typically gps from cellphones. There were also some that contained only spatiotemporal points [54], [102], [87], [84], [170], from location based queries. Other studies were unspecific on the style of the geolocation data, or the exact source of the data was not listed and so the continuity could not be interpreted.

The attacks on geolocation data followed a clear pattern. The attackers would take the available mobility traces and then find locations that could be used to identify the individual. The most common was the home/work pairings. In some cases, just the home address could be located. Geolocation data in particular poses a problem with de-identification. Though six of these data sets had some form of noise added to the data points to make them less

accurate the uniqueness of human mobility traces was still exploited. The main issue seems to be the amount of time that individuals spend at home and work creates a unique fingerprint of movement. Once those locations are discovered identification is not difficult. Even in larger cities, few people live in the same building as people they work with.

One particular data set was used in three separate studies; [35] [133] and [161], all used a data set that was released about New York Taxis. They each attacked different aspects of the data, however. [133] looked at the hashing of the taxi medallion values and discovered the algorithm used, then leveraged the limited values of the medallion to create a rainbow table and reveal all of the taxi drivers using the public information on medallion numbers. [35] took the same data set and ignored the hashing vulnerability but through analysis of the mobility traces identified many of the drivers. [161] then compared the data to paparazzi photos with the location and times of celebrities entering taxis. From that, they were able to extrapolate other trips the celebrities took based on the characteristics of the trip.

3.2.4 Differential Privacy.

For the data sets that had been protected using differential privacy, different weaknesses were exploited. For the [73] attack the data was privatized using common zero-concentrated differential privacy and Rényi differential privacy which are relaxed definitions for easier implementation. Though the researchers proved weaknesses in these implementations the main theory of differential privacy was not proven false.

For [156] and [87] the data was geological traces that were obfuscated, but this was done without consideration for the dependency between the traces. Friends and family often go to the same locations repeatedly. This could be leveraged to gain more information than the privacy guarantee should have allowed.

3.2.5 Breaking Masking

Of the attacks that were reported four performed some kind of reversal to the masking that was done to values in the data set. [84] was able to reverse the mathematical operations performed on the id numbers. The data curator had multiplied the IDs by a constant value, then subtracted another constant. Once the constants were discovered the operations could be reversed and the original IDs discovered.

As described in section 3.2.3, [133] found the type of hashing algorithm used on the medallion numbers for the taxis. This was discovered because of an error in the data input. There was an outlier of one hashed identifier appearing to have significantly more business than others. The researcher realized that a data entry error could have occurred and discovered that the associated hash string was the MD5 hash for 'o'. Leveraging the limited possibilities of original values the original values were discovered.

In these two cases, the best practice for data sanitizing was not followed. Simple reversible mathematical operations were used in place of a standard hashing or encryption algorithm for [84]. For [133] hashing without salting was performed which is not standard practice when using hashing for this purpose.

As described in section 3.2.2, [31] found a weakness in a no longer disclosed method of encryption on patient insurance numbers. This resulted in the data set being removed from the hosting website along with a description of the encryption method that was used. This example is a good model for why the method of de-identification should be available so that researchers can test the validity of the privacy claim.

For [83], bloom filter encodings were built from identifiers of a medical record system and then crypt-analysis was applied using identifiers from a voter registry. The researchers themselves took the data set and encoded then tested and broke these encodings. They also claim that the crypt-analysis, though successful, may not be practical for more realistic scenarios.

These were the only studies found that were able to break the encryption used on information within a data set. Though more attacks of this nature may be possible, with reference to the other studies found, this appears to be an attack vector with more skill required of the attacker than the average type of de-identification performed necessitates to be broken.

3.2.6 External Data Availability

Though many of the attacks set in the US used voter records to determine the identity of the individuals there are a notable number of attacks that did not require a pre-made collection of names and addresses to identify individuals. News reports on unique events, property records, social media, and others were used on data sets outside of the US and in a few cases within it.

Of the cases that specify the source of the public information used to identify the individuals, the sources included, Linkedin, undisclosed social network information, property tax registers, taxi medallion data, news articles, obituaries, death records, movie reviews, online

forum data, reverse phone number searches, ambulance, and hospital discharge records, social security death index, German employment stats.

Though the availability of these data sets may be variable as well within different countries. The only universal data availability would be newspapers and social media. It should be noted that in regards to the geolocation data attacks, depending on the intent of an attacker, public information about property owners might not be required as one could in person confirm identities once an address is disclosed.

3.3 Remarks on De-identification Attack Review

Though many of the attacks performed revealed personal information about individuals involved there is little evidence to suggest that the modern theories of de-identification are flawed. The majority of the attacks discovered were performed on data sets with removed or masked directly identifying values, but little was done beyond that to remove identities. Though there is a trend of custodians beginning to employ more modern ideas of de-identification. Of these, it was found that the implementation, not the theory was at fault for the privacy breach.

In the cases of geolocation data, browsing history [39] [8], or a homicide database [116] the information that is being studied must be altered in a more direct manner to provide security. As in the case of browser history, people will type personal information into the search bar that is easy enough to use to identify them. Or with geolocation data home addresses are where the majority of the population spends their time.

Though there will be a trade-off in utility when performing operations on data beyond removal of direct identifiers [34] it is important to maintain the public's privacy. As well the current measures and theories of de-identification it seems have yet to be rigorously tested to ensure that they provide the privacy levels promised. Increased adoption and testing of these methods would be beneficial to the entire research community and the public.

Chapter 4

Observations from Review of De-identification Attacks

In this chapter, some best practices for de-identifying different data types will be laid out. The set of best practices are based on the review of 3. These best practices are based on what has not been observed as broken at the time of this thesis' writing.

4.1 Best Practices for All Data Types

There are some aspects of de-identification that are not dependant on the type of data that is being de-identified. The actions that need to be taken on any data set intended to be de-identified are laid out in this section.

4.1.1 Suppression

When determining what data should be suppressed it is important to look at what single elements of the data set are unique to the person. There are many identifiers or personal information that custodians know to remove, such as a SIN number, credit card number, account numbers, and the like. A name is an obvious identifier that people will remove. However, an email address is likely more unique than a name. There might be many John Smith's but there is only one `johnsmith@gmail.com`. Other examples of identifiers that might be overlooked are phone numbers, physical addresses, IP addresses, and unique hardware identifiers such as MAC addresses. In many cases, these are unique to the person and so all need to be removed to protect identities [103].

When determining if a value is a direct identifier the uniqueness of the value and the ability of someone to search for that value are considered. In some cases, there is no publicly available data to compare to, for example, if a company uses a unique number to reference a user that has been randomly assigned. Releasing that information likely has no way of leading an attacker back to the person's identity. However, how that unique value was created can determine whether it is an identifier as well, see Section 4.1.3. On the other hand, values like phone numbers can be overlooked though these have been proven to be trivially re-identifiable in a study of telephone metadata [103].

When suppressing data it is important to note all of the places that the information appears in. When looking at the personal genome project the information was supposed to have direct identifiers such as names removed. However, the researchers found that because the data was supplied by the users some profiles still contained names in unexpected areas, and many of the downloadable files associated with the profile had a filename that included the name of the person [154].

A summary of the best practices of suppressing data:

1. Consider if the value is unique to the person to determine whether it is a direct identifier,

2. Ensure that none of the information seen by the attacker contains the direct identifiers you intend to suppress,

3. Consider whether the information that is being suppressed can be found or inferred from any of the values remaining in the data set,

4. Create a limit of utility to use to determine when the alterations required on the data have rendered its utility too low for the intended research.

4.1.2 Generalization

When determining how to generalize data first a set of indirect identifiers should be recognized within the data set. This can contain all of the attributes of the set. Then the privacy level that the de-identification should provide to the individuals should be set. Finally, a utility level should be set that determines how general the attribute can be before it is no longer useful for the intended research should be determined.

Though it can be difficult to know exactly what information is available to an attacker to leverage it is important to be aware of what information is available to be linked to the released data set. It has been noted that the difference in available information between regions will significantly alter the ease with which an attacker can re-identify individuals [149].

It is important to consider the distribution of values within the data set when generalizing. If the data contains individuals mostly between the ages of 30 and 50, then users outside of that range will be more unique than those inside and thus have less privacy in the data set [150, 58]. Using a *k*-anonymity measure explained in the 2.3.1 section can help to remove this difference in privacy levels, and control how much privacy a user has.

Another consideration should be made for data that has been collected over time. When generalizing time data the difference between 1 year and 2 years might be large in terms of privacy loss, but 10 years to 11 years will not have the same impact on privacy loss. Thus when generalizing time frames large time frames need to be reduced more to significantly lower the privacy loss [77].

A summary of the best practices of generalizing data:

1. All quasi-identifiers must be considered together as a set when performing risk assessment,

2. Use a *k*-anonymity measure to control the amount of generalization being done on the data and measure privacy level,

3. Generalizations should be done to a greater degree when the data is from the same population over time,

4. Reducing a data set over a time span should be proportional, the larger the time span the greater the time alteration needs to be for a significant change in risk to occur,

5. Have an awareness of the types of large organized data sets about the populace that have been made available and consider that in the risk assessment.

4.1.3 Masking

Though the reversing of masking is not a common method of re-identification, as it typically requires more skill or specialized knowledge than other methods, bad masking practices can

be a serious privacy risk. When masking is broken the original value is revealed, which could be the full name of the individual, an identifying number [84], or an account number [31].

There are different methods of masking that can be performed and it is important that when performed they are implemented correctly, used on data appropriate for their intended purpose, and any extra requirements of their standard use practice are performed. The two most common methods of masking are hashing and encryption. Though often seen as similar there are important differences in their implementations and best practices. An encryption algorithm takes the plain text and then using a key creates a ciphertext value that can then be returned to the original plain text value using either the same or another key. While a hashing algorithm will take in the plain text and output the hashed value.

There are some principles of thought that both hashing and encryption algorithms share. For one the ciphertext or hashed value that is output should not share similarities to one that came from a similar plain text value. That is to say, if the plain text is 12345, the ciphertext should be no more similar to that received from 12346 than any other. Another is Kerckhoffs' Principle, which states that "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge". That is to say, a hashing algorithm should be one in which the entire structure of the algorithm can be revealed and the attacker still cannot reverse the hash to plain text. For encryption, this means that the only secret required should be the key. For example, AES, RSA, and TripleDES are all well-known encryption algorithms the are explained in thorough detail online and are still secure. As well, a masking algorithm should also be such that observation of the output does not allow an attacker to guess any of the secret values used [84].

Hashing has unique qualities, the first of which is string length uniformity, the output will be the same size no matter the size of the original plain text that was used. For instance, SHA-256 is a hashing function that will take in plaintext of any length and always output a string of 256-bits long. This can be important as the length of output changing with the length of the input can reveal information to the attacker [84]. Hashing functions have output uniformity, meaning that the output hashed value will be the same every time the same plain text is sent through the algorithm. Hashing functions should also be mathematically infeasible to take the hash value and reverse the operations to return it to the plain text value.

The main aspect of an encryption algorithm is that it is designed to be mathematically reversible. The output ciphertext of an encryption algorithm can be returned to the original plain text through the algorithm with the correct key. This key can be either the same as was

used to encrypt the plain text (symmetric key), or a different key (public key). Encryption algorithms all have different key lengths, the Advanced Encryption Standard (AES) has typical key lengths of 128, 192, or 256-bits. The key length can affect the computational cost of the algorithm, but it also affects how long it would take an attacker to brute force attack the algorithm.

When using hashing functions best practice is to add a random salt to the value. This makes attacks like dictionaries and rainbow tables more difficult to implement against the hash. Hashing also becomes less secure when the possible input is of a limited structure. This means if the inputs are all structured the same, have the same length, are comparatively small, some digits are constant, or have a small range, then an attacker will potentially be able to brute force the hash. For example, taxi medallion numbers are all 4-6 characters with specific characters being letters and the rest numbers. Overall there are 22M possible plain text values and outputs of the hashing function or 2 minutes of computation time [133]. Another thing that should be done is looking at the data itself to ensure that there are no errors that may reveal information. If data for a few entries were imported incorrectly and are all 0 for instance, once hashed these values can create an anomaly that will reveal information to an attacker [133].

When using encryption one of the most important things to remember is that the key needs to be kept secret. Whether using public or symmetric encryption the key that is required to decrypt the ciphertext must not be stored along with the ciphertext values, or hard-coded into the system performing the encryption. Doing so would provide attackers the information that they need to reverse the masking.

A common error with masking is not using the appropriate algorithm, or not using the appropriate algorithm according to best practice. For example, if you have an ID number that needs to be encrypted, performing just any mathematical algorithm on it is not the same as performing encryption. Adding, subtracting, multiplying, or dividing the ID number by constants will change the ID value and is reversible for the data holder, but it is also reversible for the attacker. These constant values can be discovered from the analysis of the ID numbers and knowledge of the original structure of the number. For instance, the manipulations described will result in identifiers that have a different number of digits depending on the input value, comparing these different length identifiers to the original can give an attacker information they require to reverse engineer the algorithm [84]. Masking algorithms should all follow Kerckhoffs' principle.

Something that should also be considered is whether the information needs to be masked at all. Though it is necessary if the information needs to be tied back to the original individual by the data holder if all that is required is a unique identifier to be carried through the data set that does not contain private information it is more secure to use a value that is not derived from the individual at all. A completely random string or number to replace the attribute is a more secure identifier to carry through the data set than one generated by a true value. As well this would be less computationally intensive for the data holder than implementing encryption. It is important that this number not be derived from a value belonging to the original individual as seeded pseudo-random generators can be reversed depending on their implementations [30, 31].

A summary of the above best practices of masking data:

1. Masking requires mathematical operations whose secret values cannot be reverse engineered,
2. Encryption or hashing algorithms of fixed-length output should be used,
3. Weed out anomalous data or errors in the data before release,
4. When hashing is performed, random salting should also be done,
5. Hashing is not ideal for use on inputs with known limited structures,
6. Encryption keys need to be secret and secure,
7. If no reversal is required a completely random identifier should be considered for creating a perpetual identifier.

4.2 Best Practices for Demographic Data

Demographic data is not specific to a field of research. Often when collecting data for any reason there are standard values about the person that get collected as well to provide further information to researchers about the trends in the data. Examples of these types of data would include age or date of birth (dob), gender/sex, race/ethnicity, home address, name, education level, etc.

This section details information about data sets that were de-identified using the demographic data contained within them. The data sets themselves were varied in their content

otherwise, some were data sets of only demographic data, others contained health records or other information on the individuals as well.

4.2.1 Suppression

When protecting demographic data there is typically some suppression required, as some of the information will be direct identifiers. The rules of suppression explained in Section 4.1.1 apply to this type of data as well. The main idea is that any value unique to an individual should be considered as a potential direct identifier and suppressed if it can identify an individual. Names, full home address, and other things that are unique to the person need to be completely removed to protect identities [103].

4.2.2 Generalization

The indirect-identifier sets of Table 4.1 show how unique a set of values can be. Once the direct identifiers are suppressed the indirect-identifier set should be studied. These are very important for privacy, according to one study 99.98% of people in Massachusetts would be correctly re-identified in a data set containing any 15 demographic attributes [138].

The studies researched looked at populations from the US, Canada, Netherlands, and Germany, and used different types of attacks to re-identify the data. In some cases a secondary data set was used to find identities, in others, the uniqueness of a person within the set was used and if the person was completely unique this broke the privacy guarantee stated by the custodians and was considered a successful re-identification attack. Though the census information may not be available in all places, with enough knowledge of an individual through personal experience or looking for information online it could be possible to find them within the de-identified set. Some of the details of the studies of Table 4.1 are discussed in Section 4.1.2.

That is not to say that this information cannot be kept in a data set together. Generalization of these values can increase the privacy of the data set. The Canadian study into the identifiability of people in Montreal shows that altering the date of birth to the year makes less of the population unique. The same occurs when less information on someone's historical postal code record is released. From the conclusions of the study changing the date of birth to month and year of birth, as well as altering the postal code to only the first 3 digits reduces the uniqueness to a "very low value" [77].

Table 4.1: Indirect-identifier sets of demographics

Study	Year	Country	List of identifiers	Linked Data	Re-id'd
[138]	2019	USA	ZIP code, date of birth, gender, number of children	Voter registry, public information	99.8%
[149]	2017	USA	Race, gender, date of birth, education level, year they moved into their residence, home ownership status	Property tax registers, data purchased from brokers	28%
[154]	2013	USA	date of birth, ZIP code, gender	Voter list, public records website	49%
[77]	2011	CAN	date of birth, postal code	Uniqueness	98%
[77]	2011	CAN	year of birth, postal code	Uniqueness	85%
[77]	2011	CAN	date of birth, 3 char postal code, gender	Uniqueness	80%
[77]	2011	CAN	Postal code trail of 2 years	Uniqueness	17%
[77]	2011	CAN	Postal code trail of 5 years	Uniqueness	35%
[77]	2011	CAN	Postal code trail of 11 years	Uniqueness	43%
[11]	2010	USA	County, gender, date of birth, race	Voter list	60%
[11]	2010	USA	County, year of birth	Voter list	10%
[11]	2010	USA	Gender, year of birth, race	Voter list	0.25%
[11]	2010	USA	Year of birth	Voter list	0.01%
[80]	2010	NLD	4 char postal code, gender, year/month of birth	public register data	4.8%
[80]	2010	NLD	Municipality, gender, year/month of birth	public register data	0.07%
[58]	2006	USA	Gender, ZIP code, date of birth	2000 Census data	63%
[151]	2002	USA	ZIP code, date of birth, gender	Voter list	N/A
[150]	2000	USA	Gender, ZIP code, date of birth	1990 Census Data	87%
[150]	2000	USA	Gender, municipality, date of birth	1990 Census data	53%
[150]	2000	USA	Gender, county, date of birth	1990 Census data	18%
[6]	2001	GER	Income, year of birth, sex, schooling, weekly work hours, occupation, region, time employed, time unemployed, duration of previous employment, marital status, number of children, nationality	Uniqueness	69%

When looking at the indirect identifier set the external data sources need to also be considered. In the US things like publicly released census data, voter lists, data brokers, and public tax registers can provide a lot of information for an attacker. Though their availability is varied state to state or in the case of property taxes, between municipalities. In one study on air quality, the information on health data was redacted heavily according to HIPAA standard [40], however, the redacted data contained enough information to use computer inference methods to create race and gender estimations, and the data contained date of birth. Then using property tax registers and information bought from data brokers residents could be re-identified [149].

Similarly, the Personal Genome Project gathers genetic information on participants, but it was the demographic information that allowed for re-identification. The data contained date of birth, full zip code, and gender. The demographical information revealed 22% of people when

linking to voter data set and 27% of people when linking to a public records website, for a total of 49% of records [154].

In the Netherlands, research into the identifiability of hospitalization and welfare records was investigated. These records are all available upon request from the governing body that collected them. Using public register data the researchers identified 4.8% of the people in the health care set from a shortened postal code (4 values of 6), gender, and year and month of birth. The welfare data contains information about investigations of welfare fraud; using municipality, gender, year of birth, and month of birth 0.07% of people were uniquely identifiable. The total percentage of citizens identifiable to a group of 10 or less was 2.14% within the data set [80].

A summary of the above into best practices of generalization:

1. Full date of birth, postal code and gender cannot be left untouched in a data set intended to be de-identified,

2. Generalizing the date of birth to month and year of birth, as well as altering the postal code to only the first 3 digits significantly reduces the identifiability.

4.3 Best Practices for Health Data

Health data is typically information that would be in a medical record. It could contain information on prescriptions, illnesses, DNA sequences, medical procedures, or any other data that might be collected or created by doctors or nurses in a clinical setting. This data could also be from pharmacy records, insurance records, hospital discharges, or ambulance records. Some of the information collected by devices and apps that track fitness can also be considered health data. Health data is often protected by laws such as HIPAA in the United States [40] or PIPEDA in Canada [164] as well as various state or provincial level laws and standards.

If health data is released that contains information that can be re-identified there are many ways in which the individuals in the data set may be harmed. For example, personal information about the illnesses that they have had or are seeking treatments for could affect their current or future insurance rates. Previous diagnoses of mental illnesses could be used to negatively impact their current or future employment due to stigmas held by employers, as well as negative impacts on their personal lives.

Within the literature of re-identification of health data, there are a few main methods that are employed. The first is to leverage the demographical information that is included within the medical records. This is information like age, sex, race, home address, etc and the risks created by this type of data was laid out in Section 4.2 of this document. Then there is the leveraging of newspaper articles and obituaries. These attacks use the nature of reporting on motor vehicle accidents and deaths that contain the person's name to match to records and find their identity. Other attacks have used unique alleles in genomes or the unique pattern of illnesses to reveal information. There has also been a case of the masking used on health insurance numbers being broken. The breakdown of attacks is displayed in Figure 4.1. This shows that of the 18 health data re-identification attacks that were studied 33.3% were performed by leveraging the demographic information, 11.1% leveraged vulnerabilities in the masking methods used, and the other 55.6% of attacks used information unique to medical data. This is the information that will be focused on in this section.

Table 4.2 contains the information about data that was released, what was leveraged, the levels of suppression and or generalization that was applied to the information, and how that affected the identifiability of the individuals. This is based on the half of the attacks that did not solely use demographical information, those attacks were included in Table 4.1.

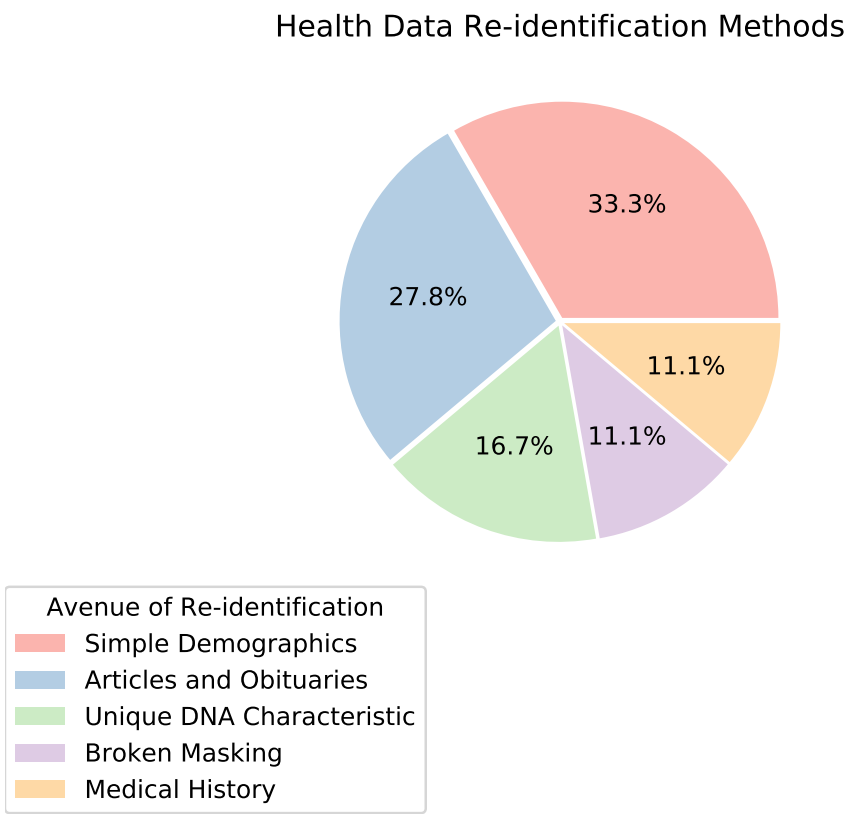


Figure 4.1: Breakdown of information used to re-identify health data

Table 4.2: Identifier sets of Health Data

Study	Year	Country	List of identifiers	Linked data	Re-id'd
[71]	2018	USA	Hospital records of vehicle accidents, year of accident, location of accident, patient age, patient sex	Newspaper articles on vehicle accidents	N/A
[110]	2018	USA	15-min aggregated physical activity data	Demographical data	94.3%
[110]	2018	USA	24-hr aggregated physical activity data	Demographical data	87%
[30]	2016	AUS	Date of hospitalization perturbed randomly by up to two weeks, year of birth, treatment	Uniqueness, public information	N/A
[30]	2016	AUS	Date of birth perturbed randomly by up to two weeks, date of child's birth perturbed randomly by up to two weeks	Uniqueness	N/A
[30]	2016	AUS	Breakdown of billing	Uniqueness	100%
[153]	2013	USA	Patient demographics, ZIP codes, diagnoses, procedures, attending physician, hospital, a summary of charges, how the bill was paid	Newspaper articles containing the word "Hospitalized"	43%
[43]	2013	CAN	Province, age at death, gender, and exact date of adverse drug event report	Obituaries	30.78%
[43]	2013	CAN	Age at death, gender, and exact date of adverse drug event report	Obituaries	5.05%
[43]	2013	CAN	Province, age at death, gender, and month and year of adverse drug event report	Obituaries	0.63%
[91]	2010	USA	Pattern of diagnosis codes	Hospital discharge data	96%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits	Hospital discharge data	96%
[91]	2010	USA	Pattern of diagnosis codes with least common 5% removed	Hospital discharge data	75%
[91]	2010	USA	Pattern of diagnosis codes with least common 15% removed	Hospital discharge data	25.6%

Table 4.2: Identifier sets of Health Data

Study	Year	Country	List of identifiers	Linked data	Re-id'd
[91]	2010	USA	Pattern of diagnosis codes with least common 25% removed	Hospital discharge data	0%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits with least common 5% removed	Hospital discharge data	70.4%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits with least common 10% removed	Hospital discharge data	48.2%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits with least common 15% removed	Hospital discharge data	16.3%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits with least common 20% removed	Hospital discharge data	0.25%
[91]	2010	USA	Pattern of diagnosis codes generalized to 3 digits with least common 25% removed	Hospital discharge data	0%
[44]	2009	CAN	Sex, age (days), postal code (3-char), admission and discharge (day/month/year)	Uniqueness	> 20%
[44]	2009	CAN	Sex, age (weeks), postal code (1-char), admission (yearly quarter), length of stay	Uniqueness	33%
[96]	2006	USA	DNA information with familial connections	Online genealogy data from obituaries	70%
[98]	2004	USA	Genetic illnesses, hospital name	Hospital discharge data, census data	98%
[152]	2003	USA	Diagnosis, inferred ZIP, drug, dosage, refill	Ambulatory data, hospital discharge data, voter list	2.3%
[97]	2000	USA	Inferred gender and illness from DNA, hospital name	Hospital discharge data	98%

4.3.1 Suppression

Often some amount of suppression is required to meet the health data protection standards in the region. For example, the HIPAA standard requires names, social security numbers, insurance plan or group numbers, medical record numbers, medical device identifiers, biometric identifiers, or any other unique identifying number, characteristic, or code, except a code to permit re-identification of the de-identified data by the data custodian to be removed from the data set [40]. Similar expectations are laid out in other legal protections for health data at the federal and provincial/state levels. The legislation of HIPAA was enacted in 1996 [40] and PIPEDA in 2000 [164], though not all US States have to follow the HIPAA standard. Based on its exemptions all of the data sets studied were released after these legislations were enacted, though only 3 directly mention HIPAA in the study.

Though all of this information has been removed from these data sets patients can still be identified. For events that are reported on by local news stations, for example, motor vehicle accidents, assaults, fires, arrests, or other unique circumstances, the year and 3 digit zip code providing a municipal location of the accident is enough to identify an article about it. News articles often contain the names of people involved especially those that were injured, thus the attacker learns their name. If the data utility does not require the location data then it is recommended to remove it, as this made finding the articles far less likely [71] [153].

In Washington state, a data set containing health information on virtually all hospitalizations occurring in the state in a given year, including patient demographics, diagnoses, procedures, attending physician, hospital, a summary of charges, and how the bill was paid, was available for $50. It did not contain patient names or addresses, only five-digit ZIP codes. Newspaper articles printed in the state for the same year that contain the word “hospitalized” often included a patient’s name and residential information and explained why the person was hospitalized, such as a vehicle accident or assault. 43% of the health records in the state could be re-identified using newspaper articles about the same events [153]. It should be noted this data was not released to HIPAA standards.

Similarly, obituary data can be used to match medical records involving patients dying. Using publicly available data-sets from statistics and health Canada on adverse drug events it was possible to match deaths to obituaries in the newspaper. Disclosing the province, age at death, gender, and exact date of the report has quite a high risk of re-identification, but the removal of the province brings down the risk significantly. By only generalizing the date of reporting to month and year and including all other variables, the risk is always low [43].

When dealing with DNA sequencing data it is important to recognize that this data alone can leak information. If the entire DNA sequence is revealed then the genetic sex of the individual is also known. Then due to the nature of genetic disorders and recent breakthroughs in research, some diseases can be directly tied to the presence of a specific gene's allele. It is possible to use this information to find patients that have these types of illnesses and match them to publicly released hospital records, using some demographic data (age, gender, generalized zip code) to confirm the match. Using hospital release records, census data, and DNA sequences released for research purposes patients with diseases such as cystic fibrosis, Huntington's disease, and sickle cell anemia were 98–100% identifiable [98] [97]. This is one example of how removing information in one area of the data set might not completely remove it from the attacker's hands.

When removing information it is important to consider whether it can be inferred from the data that has been left. For example, in prescription data the patient's zip code was removed, however, the pharmacy's zip code was included. Most people stop at a nearby pharmacy on their way home from an appointment. Thus a close approximation of the patient's zip code can be inferred about the patient from the information that was made available about the pharmacy [152].

In the table, one study found that they could use a combination of suppression and generalization to remove the possibility of their attack succeeding. Their attack relied on the unique combination of illnesses and diagnoses a single patient may have over their medical history. The problem the researchers found was that to get to 0% identifiability they were suppressing 25% of the least common diagnosis codes from all the patients, and in doing so the information was assessed to be clinically useless for the intended research [91].

A summary of the above into best practices of suppression:

1. News-worthy events require their location information to be removed or strongly generalized,

2. DNA sequences contain information that may have been intended to be removed such as gender,

3. DNA sequences containing alleles for rare genetic-based illnesses should be considered direct identifiers in many data sets.

4.3.2 Generalization

In all cases of generalization, the balance of information and privacy can be difficult to maintain. The more detailed the data the more useful, the more general the data the more private is the basics of this balance. In 2009 when studying whether the prescription data over 18 months could be released for research purposes the Children's Hospital of Eastern Ontario (CHEO), ran a study into the re-identifiability of the requested information. The original request contained the following information generalized to the indicated level if at all: sex, age (days), postal code (3 characters), and admission and discharge dates (day/month/year), as well as the drug and diagnosis information. Using a k-anonymity threshold of 5 and thus the risk of re-identification probability of 0.2 it was found that this did not provide sufficient protection of patient identities. When they considered the data usage concerns, the best solution was to raise the threshold of probability to 0.33 and provide: sex, age (weeks), length of stay (days), postal code (1 character), and admission date (quarter and year). To meet the risk threshold for some entries values had to be removed, for instance, 11.3% of the age category would still need to be suppressed [44].

When dealing with diagnosis codes in patient records it was found that in a sample of more than 96% of the records are shown to be uniquely identified by their diagnosis codes with respect to an entire population of 1.2 million patients. This was found using ICD-9 diagnosis codes when looking at the re-identifiability of disseminated EMR data. ICD-9 codes are 5 digit diagnosis codes containing three-digit disease codes, followed by two possible digits of further specification. They found that for the majority of patients the set of ICD-9 codes was unique, and thus could be used to identify a patient's record in a set containing private information. Suppressing codes that appeared in less than 5, 15, and 25% of patient's records was performed as well as generalizing the codes by removing the 2 digit specification. Both of these failed to provide sufficient de-identification [91].

When DNA data is being researched the genealogy can identify the individuals. Often with DNA data researchers are looking at inherited illness. In many cases, this means that the familial relationship between the DNA sequences are released even though identifiers are removed or generalized on the data set. By revealing the familial relationship between individuals however they can be identified. Using genealogy record sites and death records from newspapers a family structure can be built, and approximately 70% of the are unique. They can then be compared to the data revealing the family and then the individuals [96].

In a study from Australia discussed further in Section 4.3.3 researchers looked into billing records containing information on medical events, year of birth, date of event perturbed to two weeks, and other information. Researchers were able to identify Australian public figures by linking publicly available information about them to the medical records. In doing so they attempted to make the task more difficult by generalizing everyone's year of birth to 5 years but found this had little effect on their uniqueness.

A summary of the above into best practices of generalization:

1. Sex, age, postal code, and admission dates are an indirect identifier set,

2. Patient diagnosis sets are unique and should be considered an indirect identifier set,

3. The structure of a person's familial tree can also be considered an indirect identifier.

4.3.3 Perturbation

The methods by which random noise can be added to a data set vary. There are different methods of generation and not all features necessarily require noise to be added to them. In Australia, the public healthcare system released a data set containing billing information. Perpetual identifiers were used to identify the same patient across different records and other information included; year of birth, sex, medical events, codes indicating service provided or prescriptions given, the date, the location as State, the price paid, the breakdown of payment sources. The data was de-identified through suppression of some rare events and all dates were perturbed randomly by up to two weeks. Using publicly available information about well-known Australians researchers could search for mothers using their date of birth and the birth dates of their children. By querying the data, with the error in reported dates accommodated for, the individuals were shown to be unique in many cases. This indicated that the individuals were re-identifiable as they also proved there was enough information to further confirm identity within the medical record. They also found similar results from professional athletes and their known injuries, and news stories about politicians and their medical events. As well they found that the billing breakdown of payments and dates was often unique, thus private insurance companies, banks, and credit card companies, could use their own records to match to the medical records and learn the individual's medical history [30].

A summary of the above into best practices of perturbation:

1. Decreasing the precision of the data, or perturbing it statistically, makes re-identification gradually harder at a substantial cost to utility,

2. A 2-week perturbation of dates makes little impact on sparse data, increasing the perturbation has little effect.

4.3.4 Aggregation

The only health data set studied that used aggregation did so on physical activity data collected from wearable devices. The aggregation on this set was performed within the attributes, as an individual's average walking intensity for every chosen time interval was calculated and released. Researchers then used a data set containing 6 demographic variables; age, sex, educational level, annual household income, race/ethnicity, and country of birth, about the individuals in the data set to link to the aggregated walking intensity data. This matching was performed using a random forest machine learning algorithm model. It correctly matched 94.3% of adults and 87.2% of children when time intervals of 15-minutes were used. When 24-hours was the time interval it matched 87.0% of adults and 70.2% of children. This study shows that this type of physical activity data can be used to learn more about the individuals within than ever intended by the data custodian [110].

4.3.5 Access Control

A common method of access control is query control. In one case having data sets of genomic information, but only allowing yes or no responses as to whether a specific allele is in the data set. Some researchers who had access to such a data set found that because an allele's presence can be dependant on other allele's in the genome an individual's presence in the set could be discovered from queries of their single-nucleotide polymorphisms (SNPs) [159]. 5 queries to the set revealed with 95% confidence whether someone was in the set. Removing the ability to query SNPs with less than 5% frequency in populations did not affect their ability to identify presence. Due to the few required queries, limiting the number of queries to the set is ineffective as it would be necessary to limit queries below a number allowing useful analysis of the data. As well it was found that hiding parts of the genome completely caused a similar loss of usefulness to the data set to be effective at preventing the attack.

HIPAA has within its exemptions the limited data set rules. A limited data set under HIPAA is identifiable healthcare information that the HIPAA Privacy Rule permits covered entities to share information with certain entities for research purposes, public health activities, and healthcare operations without obtaining prior authorization from patients, if certain conditions are met, one condition being they signed a data use agreement that specifies: Allowable uses and disclosures, approved recipients and users of the data, an agreement that the data will not be used to contact individuals or re-identify them, require safeguards to be implemented to ensure the confidentiality of data and prevent prohibited uses and disclosures, state the discovery of improper uses and disclosures must be reported back to the covered entity, state that any subcontractors who are required to access or use the data also enter into a data use agreement and agree to comply with its requirements. With that settled a limited data set also cannot contain any of the following information: names, street addresses, or postal address information with the exception of town/city, state and zip code, phone/fax numbers, e-mail addresses, Social Security numbers, medical records numbers, health plan beneficiary numbers, other account numbers, certificate and license numbers, vehicle identifiers and serial numbers, including license plates, device identifiers, and serial numbers, URLs and IP addresses, biometric identifiers such as fingerprints, retinal scans and voiceprints, full-face photos and comparable images.

Even with all of this removed it was found that in Ohio 18.7% of the population is 1-distinct, or unique, and 59.7% are 5-distinct based on their County, Gender, Date of Birth, and Race [11]. This is compared to the risk if the data is kept under the full HIPAA protections, under Safe Harbor, 0.0003% is 1-distinct and 0.002% are 5-distinct This means that though the data is HIPAA compliant the data receivers now hold health data that can be linked back to the individual. Though the receiving entity is bound by the data use agreement to never use it for such a purpose, there is now an increased risk of another entity gaining access to the information from their data center.

A summary of the above into best practices of access control:

1. Dependencies in the attribute values need to be considered as they reveal more information than intended,

2. Access control cannot completely remove risk for the individuals in the data set.

4.4 Best Practices for Geolocation Data

Geolocation data is information based on an individual's position at a point in time. This information could be traces of their movements over time, such as from a continuous GPS connection providing constant updates of their latitude and longitude coordinates, pings of their location as they access specific services in certain locations, such as when accessing public transit, or a general location such as cell tower connections, where each tower has a range of area that the person connecting could be anywhere inside of. In all cases, the location data contains details of where someone was, and when they were there. Often from this data, a trajectory of the individual's movements can be made as they move through an area and connect to different cell towers or connect to services as they use them.

If a geolocation data set is released that contains information that can be re-identified the effect on the individuals within the set can be damaging. Attackers could learn personal information like average income [133], or habits and vices [161], and home address [36]. They could also predict where someone will go and when they will be there [170]. Attackers thus could learn detailed information about the patterns of their life as well as other private information.

Researching attacks on this type of data revealed a lot of information about the protections that were being used. For a majority of the data, nothing was done beyond removing direct identifiers to the individual's identity. Most of the data sets contained only the times, locations, and a perpetual identifier used to track traces made by the same individual through the data set.

It also revealed the methods used by the attackers to break this de-identification. The spread of the attacks and used and their course of re-identification is laid out in Figure 4.3b. There were different methods of doing this, many attackers used clustering to find likely home addresses or home and work pairs. They could also use Markov Chains to match the de-identified location patterns to known location patterns that the attacker created themselves from knowledge about the individual or from other public data that contains an identity. Some attacks were only looking at the uniqueness of the location patterns and many found that the information was unique to an individual which provides a reasonable assumption of identifiability. The only cases not discussed in the following sections are the two cases of broken masking that were studied as these are discussed in Section 4.1.3 of this document.

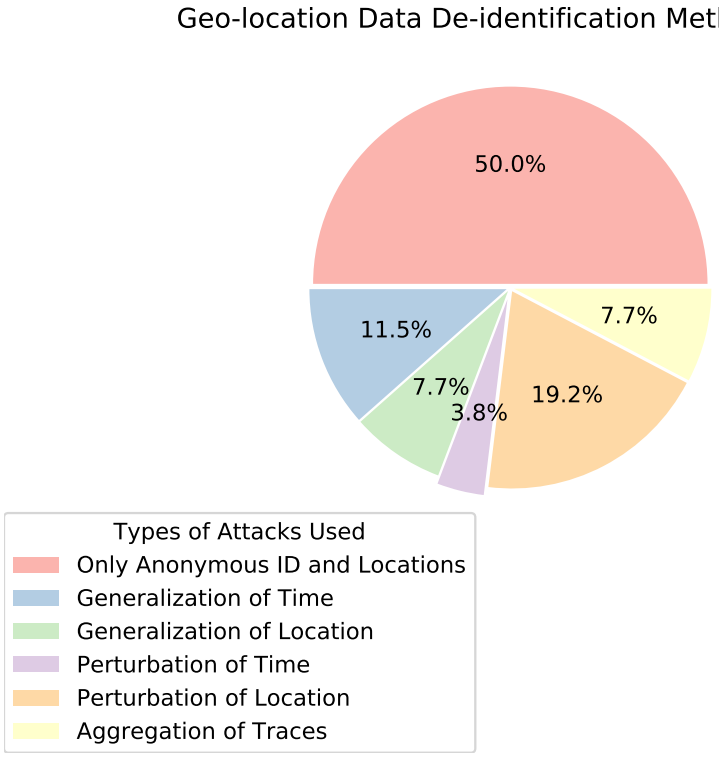
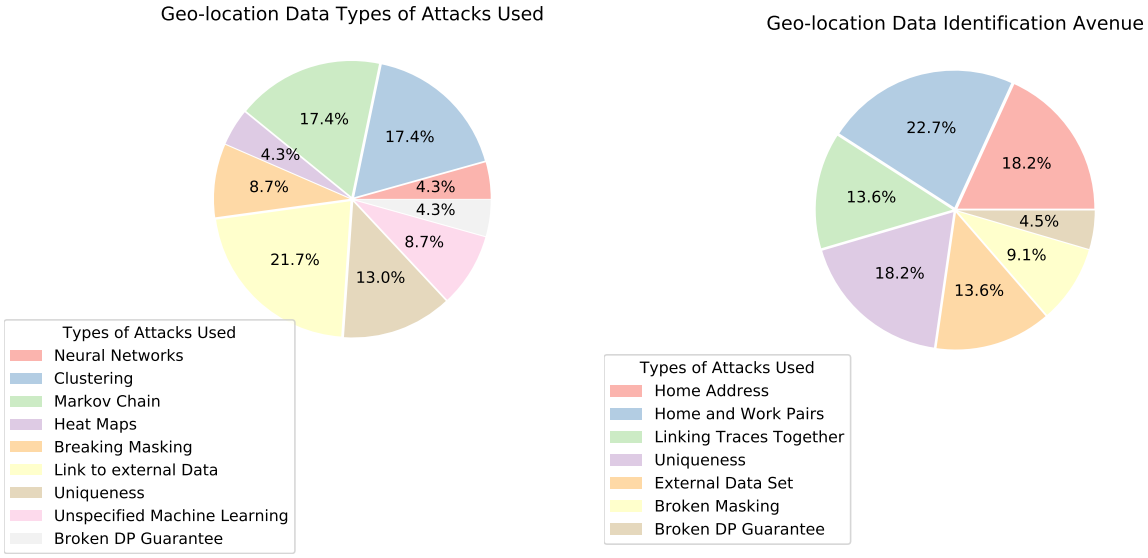


Figure 4.2: Breakdown of the de-identification used on geolocation data sets



(a) Attack methods

(b) Avenue of identification

Figure 4.3: Break down of the attack methods used on geolocation data sets

4.4.1 Suppression

With geolocation data typically suppression will include the removal of names and other known demographic information that can identify the person. Many data holders consider the times-

tamps and location as well as some form of a perpetual identifier as enough de-identification. However in multiple cases, it has been found that the location data itself when connected can leak information about the person, including home address and place of work, which together can identify an individual [59, 50, 53]. From an analysis of the census data in the US, it was found that home/work pairs at the location granularity of the census block were unique for a majority of the population, and less granularity offers more privacy [59].

In some cases, the only information released is the address of an individual. For health care, a data set with a map containing patient addresses was released with the intention of allowing analysis of illness and geographical location through the city of Boston. It was found that the data released was precisely accurate for the home address of 79% of the patients and within 14 meters for all of the addresses. From there researchers determined finding the identity of the patient was possible from an accurate home address [17].

Machine learning models can be used to match individuals between data sets. In a study performed using credit card information that contained the time and location of the transaction it was found that 4 points of an individual's time and location were enough to re-identify their trace within the data set. Overall 90% of individuals were unique with only 4 points [32].

Machine learning models can be used to create models of human behaviour, when it comes to geolocation data a neural network being fed discrete GPS locations and times a map of someone's trajectories and frequented locations can be built. Using this it is possible to use the location features of social media apps such as Twitter, Foursquare, Weibo, and ISP data to match an anonymous social media account to an identity. Researchers were able to take 21 months of twitter and 48 months of foursquare data and match accounts between the sites without looking at the content of posts or the account names. Similarly, they used 1 week of data from Weibo which used GPS coordinates, and 1 week of data from an ISP which provides the coordinate of the base station that the user was connected to at a certain time to connect individuals between the data sets. They were able to accurately match 50% of users between the two services [46]. A similar analysis can be completed using a Markov chain with a 35-45% match rate [52].

Another research group took the GPS coordinates from Twitter data and used machine learning models to create clusters of coordinates and then create a centroid of these coordinates. This centroid assumed to be the true location the person was at once the clustering accounted for the error in the coordinates. Using knowledge of human behaviour between work and home hours the models then predicted the centroid that was likely to be an individ-

ual's home and their workplace. Once this was done the contents of the tweets were searched for relevant information about the location the individual was at at the time of the tweet and the model would consider this as well. The model created by researchers was able to predict an individual's home and workplace with 92.5% accuracy [36]. Another study focused on the re-identification of 3 people in this manner and learned names, dob, occupation, family info, home and work address, several facts about their life, info about their web presence and other miscellaneous information [75].

In a case that compared bike share and jogging data, it was found that the knowledge of someone's daily routine could easily be leaked to an attacker. Many people have a daily routine that location data covering their morning or evening jog would reveal information about. As would data about their use of a bike share service to and from work, or another form of public transit. Both of these data sets provide near-continuous updates of an individual's locations and researchers could create models based on the released data of the individual's movements and predict a bike-share users location and time with 75% accuracy [102].

Using a taxi data set from New York that contained no information about the passenger's identity it was shown that it is arbitrary to re-identify a passenger in the data set if some information is known about where and when they used a taxi, or their address, as the taxi data is detailed enough to see the address that the taxi picked someone up or dropped them off at. This can reveal information previously unknown to an adversary about the trip that was taken, and the habits and behaviours of an individual. For detached homes, previous knowledge can be minimal as information about the owner can be searched through the address to reveal their identity [161]. A similar study of taxis in San Francisco and Shanghai, as well as busses in Shanghai, found that 10 pieces of external information were enough to identify a passenger in the data set. Even when the external information was inaccurate [94].

If some external information about a person's location behaviour is known then cell tower data consisting of the entry and exit time of a cell phone from each cell tower's area of coverage can be used to build a model of their location patterns and compared to this external information to find them in the data set. Using this an attacker can achieve 80% accuracy in connecting the identity to the location trace [109].

A summary of the above into best practices of suppression:

1. Home address should be considered a direct identifier for a homeowner and information that reveals a home address is an identifier,

2. The uniqueness of specific locations and patterns of movement to an individual should be considered personal information,

3. Four external locations and times are enough to uniquely identify an individual and thus require de-identification.

4.4.2 Generalization

Generalization with geolocation data can typically occur in one of two ways, either the location is generalized or the time is generalized. Potentially both of these alterations are occurring. Generalization will alter the specific latitude and longitude to cover a wider area or change the specific time to cover a greater time frame that the person may have been at that location during. Some information about the types of generalization and the identifiability of the data after this was performed is displayed in Table 4.3.

In some areas of the world, road cameras are common. In Guangzhou China road cameras along major roads take images of a vehicle's license plate and place it in a database along with the time stamp, which camera took the photo, and whether the vehicle is local or not. This amounts to a location, time, and perpetual identifier that can be used to track the vehicle along its route. This means that they can create a trajectory of movement based on these single instances in time. They attempted various time granularities but found that even with 12 hours, 5 of these records was enough to uniquely identify 90% of the individuals driving on the roads [54].

One method for generalizing the data in the time domain referred to as Promesse [135] erases user points of interest by using a speed smoothing technique, which assures that between each successive points in the obfuscated trace the distance and time difference is the same. This way someone spending a lot of time in a single location should appear similar to their location trace to somewhere they spent less time. However, with this, some methods can release information to an attacker. Though a place of interest may be harder to determine from a single trace, using a heat map to create points of interest over multiple traces is still possible and will leak enough information to identify a user [100]. This study looked at multiple data sets and re-identification types and found that the same de-identification measures did not result in the same level of de-identification of the records. Despite the structure of the data being the same in all cases the nature of the data resulting from its source, cabs, or users on a social network, can alter the effectiveness of an attack due to the patterns of movement. Cabs movements

Table 4.3: Re-identifying Geolocation Data

Study	Year	Country	Type of Data	Time	Location	Identification	Re-id'd
[54]	2019	CHN,CAN	5 Discrete vehicular locations	30 min	Exact	Uniqueness	100%
[54]	2019	CHN,CAN	5 Discrete vehicular locations	1 hr	Exact	Uniqueness	100%
[54]	2019	CHN,CAN	5 Discrete vehicular locations	3 hr	Exact	Uniqueness	98%
[54]	2019	CHN,CAN	5 Discrete vehicular locations	6 hr	Exact	Uniqueness	95%
[54]	2019	CHN,CAN	5 Discrete vehicular locations	12 hr	Exact	Uniqueness	90%
[100]	2017	USA,FRA	Discrete locations 200m apart	Exact	Exact	Uniqueness	68%
[35]	2016	USA	Taxi GPS positions	1 min	Exact	Uniqueness	91%
[35]	2016	USA	Taxi GPS positions	5 min	Exact	Uniqueness	90.3%
[35]	2016	USA	Taxi GPS positions	15 min	Exact	Uniqueness	88.6%
[35]	2016	USA	Taxi GPS positions	30 min	Exact	Uniqueness	86.7%
[35]	2016	USA	Taxi GPS positions	1 min	Census tract	Uniqueness	87.8%
[35]	2016	USA	Taxi GPS positions	5 min	Census tract	Uniqueness	83.5%
[35]	2016	USA	Taxi GPS positions	15 min	Census tract	Uniqueness	81.4%
[35]	2016	USA	Taxi GPS positions	30 min	Census tract	Uniqueness	75.5%
[35]	2016	USA	Taxi GPS positions	1 min	ZIP code	Uniqueness	84.1%
[35]	2016	USA	Taxi GPS positions	5 min	ZIP code	Uniqueness	78.4%
[35]	2016	USA	Taxi GPS positions	15 min	ZIP code	Uniqueness	68%
[35]	2016	USA	Taxi GPS positions	30 min	ZIP code	Uniqueness	54.9%
[35]	2016	USA	Taxi GPS positions	1 min	NYC neighbourhood	Uniqueness	82.5%
[35]	2016	USA	Taxi GPS positions	5 min	NYC neighbourhood	Uniqueness	70%

Table 4.3: Re-identifying Geolocation Data

Study	Year	Country	Type of Data	Time	Location	Identification	Re-id'd
[35]	2016	USA	Taxi GPS positions	15 min	NYC neighbourhood	Uniqueness	50.5%
[35]	2016	USA	Taxi GPS positions	30 min	NYC neighbourhood	Uniqueness	29.6%
[32]	2015	USA	4 Points of time and location of purchases	Exact	Exact	Uniqueness	90%
[94]	2013	USA,CHN	Taxi and bus GPS positions	1 min	0.01° coordinates	Identity matching	50%
[107]	2012	USA,CHL,BEL	4 Cell tower points covering 0.15km^2 to 15km^2	1hr	Cell tower	Uniqueness	95%
[107]	2012	USA,CHL,BEL	2 Cell tower points covering 0.15km^2 to 15km^2	1hr	Cell tower	Uniqueness	50%
[107]	2012	USA,CHL,BEL	4 Cell tower points covering 0.15km^2 to 15km^2	5hr	5 Cell towers area	Uniqueness	50%
[170]	2011	USA	Cell tower points top location	Exact	Sector	Bin-size uniqueness	372
[170]	2011	USA	Cell tower points top 2 locations	Exact	Sector	Bin-size uniqueness	2
[170]	2011	USA	Cell tower points top 3 locations	Exact	Sector	Bin-size uniqueness	1
[170]	2011	USA	Cell tower points top location	Exact	Cell	Bin-size uniqueness	967
[170]	2011	USA	Cell tower points top 2 locations	Exact	Cell	Bin-size uniqueness	9

Table 4.3: Re-identifying Geolocation Data

Study	Year	Country	Type of Data	Time	Location	Identification	Re-id'd
[170]	2011	USA	Cell tower points top 3 locations	Exact	Cell	Bin-size uniqueness	1
[170]	2011	USA	Cell tower points top location	Exact	ZIP code	Bin-size uniqueness	3125
[170]	2011	USA	Cell tower points top 2 locations	Exact	ZIP code	Bin-size uniqueness	75
[170]	2011	USA	Cell tower points top 3 locations	Exact	ZIP code	Bin-size uniqueness	2
[170]	2011	USA	Cell tower points top location	Exact	City	Bin-size uniqueness	7638
[170]	2011	USA	Cell tower points top 2 locations	Exact	City	Bin-size uniqueness	437
[170]	2011	USA	Cell tower points top 3 locations	Exact	City	Bin-size uniqueness	24
[170]	2011	USA	Cell tower points top location	Exact	County	Bin-size uniqueness	55649
[170]	2011	USA	Cell tower points top 2 locations	Exact	County	Bin-size uniqueness	15628
[170]	2011	USA	Cell tower points top 3 locations	Exact	County	Bin-size uniqueness	3407
[170]	2011	USA	Cell tower points top location	Exact	State	Bin-size uniqueness	720000
[170]	2011	USA	Cell tower points top 2 locations	Exact	State	Bin-size uniqueness	680000
[170]	2011	USA	Cell tower points top 3 locations	Exact	State	Bin-size uniqueness	460000
[59]	2009	USA	Census data home work pairs	N/A	Census block	Bin-size uniqueness	1

Table 4.3: Re-identifying Geolocation Data

Study	Year	Country	Type of Data	Time	Location	Identification	Re-id'd
[59]	2009	USA	Census data home work pairs	N/A	Census tract	Bin-size uniqueness	21
[59]	2009	USA	Census data home work pairs	N/A	County	Bin-size uniqueness	34980

have much less uniqueness than a person and so are more difficult to re-identify once de-identification is applied to the data [100].

In a study of taxi GPS traces from New York taxis, researchers tested the privacy of different generalization levels on the data by looking at how unique the traces were. This was considered a successful attack because they were also able to prove that a trace could be matched to the public medallion information which revealed the driver's identity. With location generalized to the neighbourhood and time generalized to 30-minute intervals it was still possible to identify 30% of the individuals [35].

Looking at cell phone data where the locations are generalized to the area covered by a single cell tower with 1 hr samples only 4 positions are required to uniquely identify someone. From this, it was found that statistically, traces are more unique when coarse in one dimension and fine along another, than medium-grained along both dimensions. Given four points, 40% of individuals are unique in a data set with a temporal resolution of 15 hrs or a spatial resolution of 15 antennas while 60% are unique in a data set with a temporal resolution of 7 hrs and a spatial resolution of 7 antennas. According to their analysis uniqueness decays 1/10 the power of the resolution [107].

Another study looking at cellphone data was generalizing the location starting with the sector of the cell tower's area of coverage, the cell towers area, the zip code area, city, county, and state. They base their analysis on the uniqueness of the set of most common locations starting with one location up to the top 3. From their analysis, they determined that a trace longer than 2 weeks reveals the top 2 locations of more than 50% of a population.

A summary of the above into best practices of generalization:

1. Location traces overtime should be de-identified to prevent attacks that look for important places in people's lives,

2. When generalizing locations it is important to note the uniqueness of home/work pairs extends to areas, not just exact addresses,

3. Generalizing locations so that every point is at least a specific distance apart still allows for points of interest to be found when multiple traces from an individual can be tied together,

4. Data sets from different sources should be considered new data sets when determining the required methods of de-identification as methods used on similarly structured geolocation data will not provide the same protection level,

5. Large data sets with many records for each individual require significant generalization to provide k-anonymous privacy.

4.4.3 Perturbation

Perturbation is used on coordinates to alter the exact positions slightly and create uncertainty to prevent attackers from knowing specifically where someone was, while researchers can still learn from movement patterns. Two studies of different data sets that added noise to individual geolocation traces found vulnerabilities in the implementation. They found that many of the traces were correlated, people that know each other of course often go to the same places together and immediate family will be in the same location even more often. As such models like Markov chains can be used to compare the traces and find people that are likely to know each other. Using the correlated traces and clustering the points where the individuals were likely together the real location can be found [156, 87].

Specifically, for the case of [87], the noise was applied to geolocation traces from a social networking site using a differential privacy bound and laplacian noise. By leveraging the information about relationships between individuals available on the social media website combined with the geolocation traces specific users could be inferred to know each other. Once that connection was known points on the traces where they were likely in the same location could be exploited using clustering to defeat the noise applied to the positions and learn the exact location [87].

Noise can be generated in different manners to be applied to geolocation traces. One study looking at the noise that was drawn from a planar Laplace distribution, as is the case with the Geo-Indistinguishability protocol [2], found weaknesses in this style of noise addition. Using a heatmap to create a pattern of visited locations and frequencies an attacker can still find uniqueness in the trace, and potentially reveal the user from external knowledge of their movements to link to the data set [100].

Other studies were able to use Markov chains to break perturbation applied to the location as the adversary can focus on a subset of transition probabilities. From these, the entire

transition probability matrix can be recovered. This allowed for the adversary to estimate the locations of the users and thus re-identify them [156].

A summary of the above into best practices of perturbation:

1. Noise added to the location should account for the same person being in the same place multiple times to counteract attacks based on frequency in a location,
2. The method through which the noise is generated and applied should always account for areas in which people would not be, such as the middle of a lake,
3. When noise is added to traces without considering the dependencies between the traces relationships between users can be used to learn exact locations through the noise.

4.4.4 Aggregation

Aggregating similar user's traces together is a method that can add some anonymity to the data set. Traces with similar movement patterns can be merged together to create a single trace that is then released. Essentially this creates a k-anonymous set, if 5 traces were aggregated then any trace could be at least $k = 5$ different people. However, using heat maps to match a known trace to the aggregated ones can still leak information about the user that was not previously known to the attacker [100].

4.5 Best Practices for Browsing History

Web browsing history typically consists of nothing more than accessed links and times, and occasionally a perpetual identifier used to identify the same user. There is typically nothing else in the data set yet browsing history can be used to identify people, as well as personal information about them.

4.5.1 Suppression

As with all considerations of suppression unique attributes of the data to the person need to be considered. When visiting a site like social media the URL is not always the same for all people. Sometimes when accessing features only available for your personal profile the URL is unique to you [39]. Other unique URLs might include employer websites, or google

searches containing personal information like names, ages, and locations [8]. Only 10 known searches are required to create a fingerprint of browsing behaviour that can be used to discover someone's entire browsing history from a data set of just URLs and timestamps. These unique searches should be removed from the data or altered before release.

Through social media, it is also possible to take the browser history and find someone's identity by matching their browser history to their social media feed. When sites like Twitter contain embedded links accessing the secondary website through the provided link creates a URL that indicates that this link was opened through a website such as Twitter. These types of links create a "fingerprint" that can then be matched to a Twitter feed 70% of the time

4.6 Best Practices for Call Records

Many consider telephone metadata to be without identifiers because the information contains no values that are traditionally thought of as direct identifiers, such as names [103]. However, information like locations, relationships between people, and sensitive information can be obtained from the metadata of cellphone records, all of which can contribute to re-identification.

4.6.1 Suppression

Though telephone metadata contains little information in terms of direct identifiers there is one attribute that can reveal a lot of information about an individual's identity, location, relationships, and sensitive information. Much of the telephone metadata referred to by organizations contains phone numbers. These have been found to be trivially re-identifiable using directories, and social network application programming interfaces [103].

Phone numbers in metadata can reveal not only the identity of the individual the data is from but the identity of people that they know because the call and text logs contain recipient and sender phone numbers. Phone numbers can also reveal locations from the business numbers that are called, and learn personal or sensitive information about a person based on the phone calls to businesses, doctor's offices, clinics, religious affiliations, and other organizations. Linking a phone number to a business and physical address is trivial through google places or review services such as yelp. It was found that using 10 phone calls it is possible to predict the individual's location [103].

Telephone numbers should be considered as an identifier and treated as such when de-identifying data sets. If full suppression of the phone number would reduce the utility of the

data set below utility thresholds other methods of de-identification could potentially be used to prevent identity leakage.

4.6.2 Generalization

The relationships between individuals can be inferred from the telephone metadata. From the concentration of call and text volume and length, time of day for call and text volume and length, and comparisons of whether the most called number was the most texted, and comparisons between most called number and most texted number, it is possible to determine who the individuals are in a relationship with. This is despite metadata containing none of the contents of the calls and text's [103].

4.7 Best Practices for Social Networks

Social networking data can often be released in the form of undirected graphs with nodes representing the people and edges representing their network of "friends" within the site. More information can be released along with this including usernames or geolocation positions one such case was discussed in Section 4.4 on Geolocation data.

One of the issues with most social networks is that the data that is released is available to an attacker through the nature of the social network. Even with privacy settings on the site set as high as possible some information about a person on the site is available to anyone looking for it. This would give an adversary a clear external data set to compare the information to.

4.7.1 Suppression

When these social networking graphs are released they often contain no information about the identity of the individual, and in some cases, the social network would not have a real name only a username on record. If there are names associated with the account they are not released and only the structure of the nodes representing people and the edges between them representing relationships are released.

This graph structure can be used to connect the information back to an external de-identified data source. Looking at networks from Gowalla (a social network from 2009-2012) and Google+ (a social network from 2011-2019), it was found that the uniqueness of the structure of the graph allowed for re-identification of the nodes. From the social network graphs

that were released 83.3% of users of Gowalla and 95.5% of users of Google+ could be identified. This privacy leakage would also get worse the more nodes and edges that a network had [76].

A similar attack was done with Twitter and Flickr accounts. By looking at the structure of the networks they were able to match anonymous accounts between the two social networks. The algorithm used found a few matching node pairs between the networks and was able to use these to expand to further nodes and identify 30.8% of the nodes that appeared in both networks [113].

Suppression of the structure does not provide enough privacy for a balance with the utility of the data. For instance, if the degree of a node (the number of edges, meaning, in this case, direct relationships to other nodes) is unique, then someone could be re-identifiable from that information. Applying k-anonymity principles to the degree of the nodes so that $k - 1$ nodes all have the same degree is a method that has been proposed to de-identify these graphs [88]. When one such method was implemented and attacked it was found that an attack that focuses on the structure of the network was still successful, as there was an overlap of 58% of the edges in the network with the external data. Higher k values would have required the removal of more data that would make the entire data set far less useful [99].

4.7.2 Perturbation

An interesting method of perturbing the information in a social network graph was proposed that involves adding and deleting edges of the graph at random [169]. The idea is to create noise in the structure of the graph and prevent an attacker from being able to use this structural information to match the de-identified node back to the true node and de-identify the user. However, a structural based re-identification attack was successful for 61% of the nodes in this data until while the added noise was held to an acceptable level of error [99].

4.7.3 Aggregation

A different way of implementing a k-anonymous measure of privacy to a social network graph is to group similar graph nodes into clusters with a minimum size constraint [160]. This was implemented and then attacked using structure-based methods. Despite the de-identification there remained 63% overlap with the external data when the method was implemented on a network at acceptable utility levels [99].

4.8 Best Practices for Billing Information

Billing information includes details about any products or services that were paid for by one group to another. This typically involves a breakdown of the amount paid, and how it was paid, any tip that might have been added on top of that, and potentially details about what was purchased.

4.8.1 Suppression

With billing information, there is always some suppression of data, credit card information, bank account numbers, and other data that someone could use to fraudulently make transactions must be removed from the data before release.

In a case of credit card data being released that contained times, locations, and the amount paid it was found that 4 locations known by an adversary could identify 90% of the individuals in the data set, this is discussed in Section 4.4, and that the adversary knowing the price of the transaction increased the re-identification risk by 22%. Their attack was performed using machine learning methods applied to the data and looking to match records within the data set to their external data [32].

4.8.2 Perturbation

When applying noise to a data set the current common method involves some form of differential privacy. In a study, three different definitions of differential privacy were implemented on a data set containing customer purchase records corresponding to 100 frequently purchased items. The differential privacy implementation that were tested were naïve composition, advanced composition[38], zero concentrated differential privacy [19], and Rényi differential privacy [105]. The study focused on using the data for machine learning purposes and comparing the utility of the data to the chosen privacy budget ϵ. They concluded that for the machine learning models to create accurate predictions of the data the privacy budget has to be set far too high to provide any protection from a realistic adversary [73].

4.9 Conclusions from the De-identification Attack Review

De-identification is the process of lowering the probability of information being used to identify an individual or a unique pattern in the data set. There are six general ways that this can be done, and their effectiveness depends on the data type, the specific data set, and the external data available to an attacker. The effectiveness of a de-identification method depends on the data type it is being implemented on and the contents of the data set. Key considerations need to be made based on the type of external data available to an attacker and the uniqueness of the values or set of values in the data set.

Though many of the attacks investigated revealed personal information about individuals involved, there is little evidence to suggest that the modern theorems of de-identification are flawed. The majority of data sets had the direct identifiers removed or masked, but little was done beyond that to remove identities. Though there does appear to be an increase of custodians beginning to employ other methods of de-identification in recent years, as all studies involving obfuscation of data are from after 2010, with 10 of the 12 being from the last 3 years (2016) [54, 73, 148, 156, 159, 51, 100, 155, 87, 30, 94, 53]. These found that the implementation, not the theory was at fault for the privacy breach.

Though there will be a trade-off in utility when performing operations on data beyond removal of direct identifiers [34] it is important to maintain the public's privacy. Data custodians need to be up to date on de-identification methods and implementations, as well as documented failures of them. Increased adoption and testing of de-identification methods would be beneficial to the entire research community and the public. Testing these methods on real-world data sets is the best way to determine what methods and standards work and which do not. Testing will guide best practices to be better and give data custodians more information they can use when considering the de-identification methods, standards, and processes their data will require.

Part II

Security and Privacy of Patient Contact Tracing

Chapter 5

Background on Contact Tracing

In December of 2019, the office of the World Health Organization (WHO) in the People's Republic of China received a media statement from the Wuhan Municipal Health Commission on cases of 'viral pneumonia' in the city of Wuhan. On January 30^{th} 2020 the WHO Director-General declared the novel coronavirus outbreak a public health emergency of international concern (PHEIC), the WHO's highest level of alarm. On March 11^{th} due to the concerning levels of spread and severity of symptoms the WHO made the assessment that the coronavirus disease of 2019 (COVID-19) could be characterized as a pandemic [131]. The Director-General said that "all countries can still change the course of this pandemic" if they "detect, test, treat, isolate, trace, and mobilize their people in the response" [55].

On March 20^{th} 2020 the government of Singapore released a mobile app called Trace Together. This app was based on the open trace code base which is an open-source implementation of the BlueTrace protocol that they also released. This was the first national Bluetooth contact tracing app in the world [127]. The intention of a tracing app is to facilitate the process of notifying all of the people that a contagious person has come into contact with that they may have been exposed to the virus.

5.1 History of Contact Tracing

Contact tracing is not a new idea. 500 years ago a physician used tracing methods to track the bubonic plague [28]. Methods of tracking how an illness spread has also been used with syphilis and yellow fever (though it was later found yellow fever was not spread through human transmission). A hospital duties book from Nuremberg Germany compiled between 1500 and

1700 contains a list of questions every patient was asked relating to how, when, where and, if possible, from whom the patient had contracted their illness [28]. In more modern medicine practices contact tracing has been used widely when patients are diagnosed with a sexually transmitted illness (STI), though this is often referred to as partner notification or partner services [12].

Originally doctors studying the spread of illness focused on disease tracking. Doctors sought to track what was the great pox, and would later be known as Syphilis, in the early 1500s by following and tracking the disease using contemporary histories including the Journals of Christopher Columbus. They could track the spread of the disease in this manner through the Americas to Barcelona then through soldiers and others to Italy with the Siege of Naples in 1495. This is what we might now call a "super spreader" event as the siege itself and then the dispersal of mercenary soldiers after. Then the disease continued to spread into eastern Europe likely due to Venetian Commerce. [28]

The earliest record of a doctor tracking whom a person came into contact with or where they went was during the Italian Bubonic plague outbreak of 1576. Doctor Andrea Gratiolo aimed to prove that a woman who had travelled between two cities had not been the cause of the plague's spread. The doctor argued that if the woman had been the cause of the spread, then the passengers in the boat she had travelled in because of the tight conditions would have caught the illness, but they had not. Neither had anyone in the woman's household. [28]

Now contact tracing has changed from tracking the spread of disease to preventing the spread of disease. Using it as a tool doctors aim to prevent further spread and catch cases sooner. Hoping that the sooner they know someone has the disease the sooner they can begin treatment and the more likely they are to recover or the better chances that they have [166].

A common form of contact tracing that people may be familiar with because of examples used in popular culture is what is known as "Partner notification responsibility" in Ontario. This is where health professionals have a responsibility to notify the sexual partners of a patient that has been diagnosed with an STI. This form of contact tracing typically consists of the patient themselves listing the names and contact information of anyone they could have transmitted the illness to and either the Physician or patient contacting them that they should get tested. [130]

There are other examples of modern contact tracing before now. During the 2014 Ebola pandemic in Guinea, it was used as a tool to contain the virus [139]. It was used during the H1N1 pandemic [142], as well as other modern pandemics. Manual contact tracing is a

three-step process. First, ask the infected person about their activities and the people around them during the contagious period of the illness. This typically consists of family, colleagues, friends, and health care providers. Second efforts are made to identify these contacts and inform them. This would be telling them about their contact status, what actions to take, and symptoms to watch for. In some cases, isolation or quarantine might be required, either in their home or at a hospital. Then lastly there is follow-up. Regular checks should be conducted to monitor for symptoms and test for infection [166].

5.1.1 Privacy Concerns of Traditional Contact Tracing

There have been privacy concerns with contact tracing since it became a more common practice. The information about the patient and the people they were in contact with is both private health information and personal information. The contact tracer requires the name and contact info for all parties in order to perform their task, as well they likely will learn or need to know what the relationship between parties are, how often they are with each other, and where they meet. They could even know where they live, as in some places the contact tracers will go to the address if they cannot contact someone by phone [132].

Some of the concerns can be mitigated by procedures similar to what some call centers implement and releasing to the public statements about these procedures. The privacy concerns themselves are often best written as questions about what the procedures for the tracing team are. Since the contact tracer themselves now has the information, are there safeguards in place to make sure that they are not keeping the personal contact info, etc. from the patient or the people that they have to contact? Are the calls recorded? Are new hires subject to background checks? What kind of database is any information being entered into? What happens to notes that the tracer makes while working? How is the data secured? How long is data retained and under what policies? If there is a data breach where is the liability? [56] Providing these answers would create a clearer picture of the privacy protections available through manual contact tracing.

5.2 Purpose of Contact Tracing

It is important that the purpose of why something is being designed is kept in mind during the process. When a group sets out to do something, if the main goal is not always at the forefront of the design it can result in an ineffective solution. The point is more than we manually contact

trace, so why not try and automate it. If there is no evidence that contact tracing helps, then perhaps we should not be making technology-based versions, or something fundamental in what contact tracing is, needs to change. The first thing to look at is whether there is evidence that contact tracing aids doctors in stopping the spread of the disease and treating patients. Then what the defined goal of the design is, as there are different ways to implement a system.

5.2.1 Benefits of Contact Tracing for the Populace

There are many reasons that doctors and researchers may be interested in contact tracing. For one people who are in close contact with someone infected with an illness that is or possibly is human-to-human transmissible are at a higher risk of becoming infected and then infecting others. Thus finding and monitoring or testing these people can allow medical professionals to get them the care they need faster and prevent further spread of the virus. [166]

It has been proven through studies of partner notification that allowing the partners to seek care early if they have an infection, can prevent them from spreading the disease. Informing them of the possibility as early as possible enables them to take preventative actions. It also reduces incidents of serious complications from these illnesses. [130]

It was found that during the 2014 Ebola outbreak in Guinea contact tracing was a significant tool for epidemiologists in containing the infection. The use of contact tracing reduced the time it took to detect and treat cases and significantly minimized the risk of transmission to subsequent individuals. However, its success is determined by the level of trust between the community and the public health system. [139]

5.2.2 The Goals of Contact Tracing

Though contact tracing uses have been discussed, as well as how it can benefit the populace it is still important to discuss what the goals of creating a contact tracing program are. As modern contact tracing based on technology brings with it more possibilities of what contact tracing can do.

There are three separate use models that appear commonly in many of the proposed and implemented contact tracing schemes. [137]

1. Inform the health authorities about who might need to be tested or quarantined

2. Serve as a digital permission slip to access various services

3. Inform people that they might have been infected so they can consider getting tested and isolating

The various applications that will be discussed all are designed to meet at least one of these goals. Some have multiple functions and can meet more than one goal. However, there are some ways in which the goals are counter to each other. In the first, the health authorities need to know who is at risk so that they can contact them, wherein the third the health authorities may not know who needs tests. In the second model, the digital permission slip reveals to those it is shown someone's risk status, thus revealing information to someone if the individual wishes to access the service. Depending on what the service is this can essentially remove the consent from the release of this data if the service is essential like a grocery store. [137]

5.3 Digital Contact Tracing

There are many ways that modern technology can be leveraged to perform contact tracing. It can be as simple as using a calendar app to determine your locations over the last two weeks, or as advanced as using the GPS from your phone to know exactly where you were at every minute of the day. Many different technologies have been suggested to assist with manual contact tracing. Including GPS [123], Bluetooth [127, 158, 162, 67], speaker and microphones at ultrasonic frequencies [68], QR codes [115] and others.

In Chapter 6 the full protocols will be explained for these technologies. Here the basic idea will be covered for some of the contact tracing methods that have been proposed or are in use.

Bluetooth Based Contact Tracing Methods

An app broadcasts an anonymous identifier using Bluetooth and other devices running the app scan for this. When the devices detect each other they exchange identifiers and log that and the signal strength. If someone then tests positive for COVID-19 they release information that allows the system to alert the people they were close enough to for long enough to be at risk. This is typically 2 meters for 15 minutes. [127]

GPS Based Contact Tracing Methods

An app logs the GPS location data of the phone and either broadcasts it live to a server or stores it. When someone tests positive the places they have been are identified. Then either

other people that were there at the same time are notified, or a publicly available map is updated and people compare to see if they were in an at-risk location. [93]

Contact Tracing with Heat Maps

These are typically hosted on a website and inform users if areas they are, have been, or plan to go are at-risk areas. This can be connected to a GPS based system, or be based on patients recalling where they have been. It could even be based on where patients live. There is a lot of variance on what exactly the map is showing. [26]

Audio Based Contact Tracing Methods

These can be used in conjunction with other methods, an app would use the microphone and speaker of the phone to determine the distance to other devices at the time that an interaction is logged. The idea being that it is more accurate as sound does not travel well through walls and so the app would have fewer false positives. [68]

Contact Tracing with QR Codes

QR codes are seen to be used in two different ways. The first is location logging. A restaurant or store will have a QR code that they display. Upon entering the establishment a user takes a picture scan of the QR code in the app and the location and time are logged. Then if someone later tests positive the location can be informed and anyone who does not have the ID exchange app could still look up the list of at-risk locations and times and determine if they should get tested.

The other way in which QR codes are used is they are generated in the app based on a user's risk level. This risk level is determined in a variety of ways. Things like where they have been, where they live, how many people they have been around, whether any of those people have reported testing positive etc. Then when someone tries to enter an establishment like a store, or public transit a worker scans their QR code and allows them entry if they are below the risk threshold.

5.3.1 Privacy Concerns of Digital Contact Tracing

This is all-important for the mitigation of the spread of the virus within a community. However, there are many ways in which it can become a threat to the privacy of individuals within that

community. When this tracing is performed manually it is limited by the nature of human memory. A human has to remember every place they were for two weeks, and everyone who was with them. A simple task for people you are regularly around, that becomes much harder when you consider public spaces where there could be many people nearby that you do not personally know. The introduction of technology is intended to fill the gap of human limitations and perfectly recall every place and every person whether you know them or not. An admirable goal for the fighting of a disease, a potential threat to the privacy of the individual.

Making available not only everywhere you have been, but every person you have interacted with as well as how long you were with them, is information that can be used to determine who you are even if your name is not connected to the data. From the location data, someone can determine your home address or work address based on how often you are there, how long you spend there, and the hours that you are there during [36]. Similar to how home and work can be determined friends and family could be determined by the number of times that they appear nearby, coworkers could be determined based on hours spent together, romantic partners as well. Then there is the addition of whether the data is being transmitted constantly or stored on the phone and uploaded. Having the ability to know where someone is at all times accurately rather than probabilistic guessing is powerful data with the potential for abuse. There is also the potential for security threats that could allow someone access to the system that they should not have, or other forms of attack that undermine the trust in the system as well as prevent health care providers from being able to use the system effectively.

Chapter 6

Contact Tracing Schemes

Turning contact tracing into a more automated digital system can take many forms. Every team has slightly different ideas of what needs to be done and every country can have different requirements for societal or legal reasons. The two main competing technologies are Bluetooth and GPS. There are also two competing data control systems, the centralized and decentralized systems. The differences of these methods and their combinations will be described in this section along with the protocols that have been developed for the apps that will be the focus of this thesis.

6.1 Bluetooth Contact Tracing Systems

The Bluetooth contact tracing technology focuses on who and for how long someone is in contact with another. The main idea of the system is that the app scans the Bluetooth wavelength looking for other apps. When it finds one the two devices perform a Bluetooth handshake, exchange strings that act as identifiers, and then disconnect from each other. Information like signal strength at the time of connection is also passed. The signal strength is used to measure the distance between the two devices. Then at a later time if someone tests positive the identifiers are compared to determine who might be at risk of having caught the illness. The people identified from the comparison are contacted and informed of the risk so they might isolate and test as they would with manual contact tracing methods [127].

Modern devices use the Bluetooth low energy (BLE) protocol. In BLE parlance there are two roles during communication. A device is either has the peripheral or central role. In every handshake, one device takes the central and the other the peripheral role. In the first stage of

communicating, that is finding the other device, the peripheral is advertising services, while the central is scanning advertisements. A service is a collection of data, such as characteristics, that the central device can read. Once the central finds the peripheral advertising what it is looking for it can move to the next step of the handshake which is discovering the peripheral's characteristics and reading the value. For contact tracing the characteristic is information like the identifying string and signal strength. The central reads this information and writes it into its own memory, then it writes back its own characteristic value to the peripheral. This is important as the peripheral has no write privileges during this exchange. This write-back allows both devices to store the other's identifying string and signal strength [127].

It should be noted that on all of the android applications that use a Bluetooth based method location services need to be enabled for the app. This is because the android software bundles the Bluetooth permissions in with the location services permissions. For the app to access the Bluetooth services continually as it does these permissions are required, though the application does not use the GPS based location services.

6.1.1 Centralized Bluetooth Contact Tracing Systems

The basic framework of the centralized data approach is as follows and can be seen in figure 6.1. Users download the app and register with the health authority. The health authority server passes them a list of IDs that they will rotate through for a period of time, lets say every day it gives a list of IDs that are incremented through every 15 minutes. When two devices are close enough they exchange the ID currently in use. User A stores User B's ID and B stores A in a "contact list". Then if A tests positive they authenticate this information with the health authority and upload their contact list to the server. On the server, the list of contact IDs is processed. The infectious period of the positive user is determined and any IDs on the list during that time are determined and their corresponding user identified. In this case user B will get a notification [127].

What makes the system "centralized" is that the processing of contacts and risk levels is performed on the central server. This is not the case in the "decentralized" system.

BlueTrace

BlueTrace is the protocol that was developed in Singapore by the department of ___. There is an open-source code base called Open Trace that was created to allow others to more easily

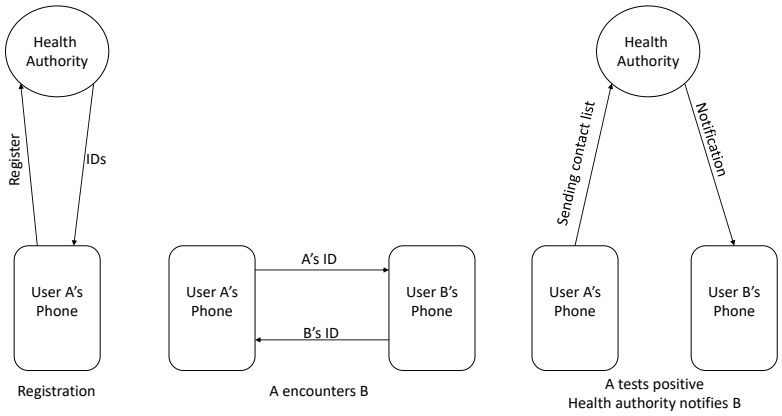


Figure 6.1: Visualization of the basic design of the centralized system [137]

adopt this protocol and the TraceTogether contact tracing app released in Singapore is based on that code base [127].

The BlueTrace protocol starts with the registration. A user downloads the app then the back-end service generates a unique randomized user ID that it associates with the phone number the user provides. The purpose of the phone number is to contact the user in case of exposure. Then the back-end server generates the first set of temporary IDs (TempIDs) that will be used. A TempID is comprised of the user ID, the use start time, the expiry time, a random initialization vector (IV), and an authentication tag. The user ID, use start time, and expiry time are all encrypted using AES-256-GCM encryption. Then the initialization vector and authentication tag are concatenated onto the string resulting from the encryption. Then the entire thing is base64 encoded. One single tempID is 84 bytes in length. The start and expiry time refer to the time block that the device will be broadcasting this ID for [127].

The tempIDs are designed with some security in mind. They have a set length of time they should be broadcast for to prevent a permanent identifier from being broadcast publicly. A public broadcast could allow an attacker or the health authority to track an individual. It will also help to prevent replay attacks. An attacker could read an ID and then broadcast it themselves somewhere else pretending to be another user. The time limit means this would only work for the valid period of that tempID as the attacker would have to keep grabbing the user's tempID to continue the attack. BlueTrace recommends a lifespan of 15 minutes for a single tempID. The protocol also recommends that the IDs be sent in batches that are forward dated to the user to ensure that the same tempID is not broadcast longer than the set time [127].

The entire message that is recorded at an encounter is more than just the tempID. The message stored in the contact list is the time stamp, version of software, tempID, organization, sending device model, receiving device model, signal strength. The organization refers to the authority under which the app is operating, in Singapore, this is the Singapore government. For the Australian implementation, this is the Australian government. The sending device model is the make and model of the device that was connected to. That is user A would have the make and model of user B's phone listed. The receiving device model is the user's own device's make and model. The reason for this is that the developers discovered that different models of cellphones had different BLE signal strengths. Thus the only way to adequately measure distance was for them to test many cellphones and create a database. When determining if a contact is at risk the back-end server references the signal strength database to determine how far apart the devices were. [157]

To ensure that as many devices in an area as possible are contacted instead of the same device over and over a blacklist is created in the app. The blacklist contains recently seen devices. Any devices on the list will not be contacted again until they are removed from the list. According to the protocol a device should be blacklisted for between one and two scanning cycles [127].

The contact list that the device creates is only held for a limited amount of time. On a rolling window, the list should only have contacts from the 21 days prior to the current date. Any older than that are completely deleted. The reasoning is that only the contacts during the patient's infectious window are relevant to contact tracing [127].

The back-end server is where the risk assessment is performed. A positive user will upload their contact list to the health authority server. This upload requires an authorization code that will be provided to the individual by the health authority. When the code is entered in the app the server authenticates it, then provides a valid token allowing the contact list to be uploaded. The health authority then decrypts the tempID of each contact, obtaining the user ID and valid period for that tempID. It then finds the tempID list of that user and verifies that the encounter falls within the validity period of the tempID. Then the risk assessment is done based on the disease's epidemiological parameters, the length of exposure, and distance. Length of exposure is measured by the length of a continuous cluster of encounters. Distance is measured by the received signal strength. Using these parameters the health authority determines which users need to be informed of their possible exposure [127].

The protocol recommends that a manual interview is still performed with the patient. This can be used to adjust the information that has been collected by the app [127].

ROBERT

The ROBERT (ROBust and privacy-presERving proximity Tracing) scheme is the result of collaborative work between Inria (led by the PRIVATICS team) and Fraunhofer AISEC. It was first implemented in the contact tracing app released by the government of France called StopCovid, then in the update of that app now called TousAntiCovid.

As is the standard order of operations the first stage of the protocol is the registration, referred to in the ROBERT protocol as initialization. Once a user has downloaded the app then registers with the server. The server generates a permanent identifier (user ID) and several Ephemeral Bluetooth Identifiers (EBIDs). The back-end maintains a table, the ID table, that keeps an entry for each registered ID. The stored information is not associated with a particular user (no personal information is stored in the ID table). [158]

The ID table contains the following information for every user A. The authentication key for user A. The authentication key is of some length greater than 128 bits and shared with the app of A to authenticate messages. The encryption key for user A. The encryption key is also of some length greater than 128 bits and shared with the app of A to encrypt information sent from the server to the app. The permanent identifier for A. The permanent identifier is a 40-bit identifier that is unique for each registered app and generated randomly without replacement to avoid collisions, this is not shared with the app. A notified flag for user A. This flag indicates if the associated user has already been notified to be at risk of exposure (“true”) or not (“false”). It is initialized with the value “false”. Once set to “true”, the user is not allowed to make any additional status request. The flag can be reset if the user can prove that they are not at risk anymore. A status request epoch. The epoch is a 24-bit value indicating the last epoch that user A sent a status request (detailed below). A list of exposed epochs. Each time one of A’s EBID appears in the proximity list of an infected user, the epoch j when the encounter happened between the infected user and A is added to this list. [158]

During the registration, a user will have to pass a proof-of-work system such as a CAPTCHA, which the system will use to avoid automatic bot registrations. The app will then establish with the server the duration of an epoch, current epoch time, and starting time of the next epoch. The epoch information is used to synchronize with the user. The app will then receive the

authorization and encryption keys from the server by means of a key establishment procedure. Finally, the app receives an initial list of ephemeral IDs and country code pairs. [158]

The generation of the ephemeral Bluetooth identifiers is performed on the server. Every set amount of epochs the app will connect to receive a list of EBIDs and encrypted country codes (ECCs) pairs. When the server receives the request it generates the pairs. The country code is used to determine which country the app that is being communicated with is from. Having this allows the server to identify when a contact is using an app developed by another country and potentially pass at risk IDs to the server for that country to notify their user. [158]

An EBID is a 64-bit identifier generated for an epoch i using a 64-bit block cipher as shown in equation 6.1. The ECC is an 8-bit code that can be decrypted by back-end servers that have federation agreements providing them the information required to decrypt another country's country code. The method for this encryption is detailed in equation 6.2. The country code is encrypted using AES in Output feedback (OFB) mode. In this form, the federation key is used to encrypt the EBID that has been padded with zeros. Then the most significant bits (MSB) are taken and XOR'd with the country code (CC). The list is encrypted before being passed to the app using AES-GCM with the app encryption key and a random 96-bit initialization vector. [158]

$$EBID_i = ENC(K_s, i|ID) \tag{6.1}$$

Where ENC is a 64-bit block cipher, K_s is the server encryption key, and ID is the permanent ID of the user

$$ECC_i = MSB(AES(K_G, EBID_i|0^{64})) \oplus CC \tag{6.2}$$

AES using Output feedback (OFB) mode is used where the key K_G is the federation key, CC is the country code, and the EBID is used as the initialization vector. Then the most significant bits are taken

It should be noted that by using the padded EBID as the initialization vector in the AES-OFB encryption they are using an initialization vector that is shorter than the 128-bit required. As the randomness of the IV is now relegated to the first 64 bits. Assuming that K_G is the same for every user, which is implied, it makes it cryptographically possible for an attacker to learn two individuals' country codes. With a 64-bit randomly pulled number using the birthday paradox after 2^32 random draws, it is likely there will be a collision. Thus if an attacker were to

set up Bluetooth receivers in a busy area, for example, an airport, after they have collected 2^32 or approximately 4 billion EBID and ECC pairs they will have collected two EBIDs that are the same. If the country codes are different the attacker can XOR them to get the unencrypted country code. This is made slightly more possible by the fact the EBID is a temporary value that gets regularly rotated and so many are being generated. If the attacker knows the country codes then they will learn the country of the individuals though be unable to know with that information alone which individual is from which country. The country codes would have to be known but as they are just an 8-bit number representing a country there is no reason to assume they are secret.

Then the app goes into the proximity discovery stage. Once the app has a set of EBIDs to broadcast and permissions on the device it begins to do so. When it detects other devices over Bluetooth it exchanges encounter messages with them. These messages contain the encountered app's EBID. The app collects the messages received and stores them, together with the time of reception and possibly other information such as the strength of the Bluetooth signal or the user's speed into a local list of the application, the LocalProximityList. This is equivalent to the BlueTrace encounter message and contact list. [158] [158]

The message that is broadcast between devices consists of the ECC, EBID, a 16-bit timestamp, and a 40-bit message authentication code (MAC). The entire message has a total length of 128-bits. The layout of the message is seen in equation 6.3. The MAC is created by using HMAC-SHA256 with the user authentication key as the key and an 8-bit prefix '01' concatenated onto the ECC, EBID, and 16-bit timestamp as the IV as described in equation 6.3. The MAC is then truncated to 40-bits. [158]

$$M_i = ECC_i|EBID_i|Time|MAC_i \tag{6.3}$$

$$MAC_i = HMAC - SHA256(K_A, c_1|M_A) \tag{6.4}$$

Where K_A is the users authentication key with the server, c_1 is the 8-bit prefix '01', and M_A is the ECC, EBID, and 16-bit timestamp.

If a user tests positive for Covid-19 they initiate the declaration stage. With the user's consent and authorization from medical services, the user's LocalProximityList is uploaded to the server. Only the contacts recorded during the infectious period are uploaded. ROBERT specifies that the LocalProximityList should be removed of any information that could tie it to

the infected user. The server then goes through the messages, verifies the times and MAC of the message, and determines if that EBID is at risk. EBIDs determined at risk are added to the exposed ID list. After the message is processed it is deleted from the server. [158]

The ROBERT protocol suggests different solutions to the possibility of the server being able to rebuild the social graph of the infected individual from their LocalProximityList. This kind of attack was described in section 4.7 and examples of this attack being performed were detailed. To break the link between any two EBIDs in the list multiple solutions are suggested. The first solution is that every item in the list is sent individually using a mixnet. The second is that the LocalProximityList is uploaded to a trusted server like that of a hospital or health organization that mixes all the lists of all the infected users together. The back-end server would then access this mixed list. The third solution is that the back-end server is equipped with a secure hardware component that processes the uploads of the LocalProximityList. Then the back-end server only has access to the exposed entries via the secured hardware. [158]

Every initialized app periodically sends exposure status requests. The app queries the server by sending its EBIDs. The server then checks if these EBIDs have been flagged as exposed. If any EBIDs have been flagged the server computes a risk score from information like how many times it was flagged, the exposure duration, or the user's speed/acceleration during the contact. If the risk is larger than a given threshold, the ROBERT protocol does not specify the threshold, then the server returns the query with a '1' bit meaning at risk of exposure. If there is no risk the server returns a '0' bit to the app. When the app receives a '1' a notification is displayed to the user. [158]

The query that the app sends consists of the EBID, its valid epoch of the EBID, the 32-bit timestamp of the transmission time, and a MAC. The MAC is again created using HMAC-SHA256 using the authentication key of the user as the key and an 8-bit prefix '02' concatenated onto the EBID, epoch, and time stamp as the IV, as displayed in equation 6.5. [158]

$$Request_i = EBID_i|i|Time|MAC_i \tag{6.5}$$

$$MAC_i = HMAC - SHA256(K_A, c_2|EBID_i|i|Time) \tag{6.6}$$

Where K_A is the user's authentication key with the server, and c_2 is the 8-bit prefix '02'

When the server receives the query it parses out the information. Then verifies that the time is close to its current time. Then decrypts the EBID to get the user ID which is used to get

the ID table of the user which is needed to get the authentication key of the user, check the user notification flag, the status request epoch, and the list of exposed EBIDs. Then the server verifies the MAC, an incorrect MAC results in the request being silently rejected. The server verifies that the user notification flag is currently false, if the flag is true the request is silently rejected. The server also verifies that the last request was at least T number of epochs ago, T being some value, if the last request was too recent the request is silently rejected. In the case of a valid request, the server updates the status request epoch of the user with the current epoch. Computes a risk score value derived from the list of exposed EBIDs. Replies with either a 1 or 0 if the user is at risk, or not respectively. [158]

6.1.2 Decentralized Bluetooth Contact Tracing Systems

The basic framework of the decentralized data approach is as follows and can be seen in figure 6.2. Users download the app and the app generates a list of IDs. The IDs will be used for a short period of time each, let us say every day it creates a list of IDs that are incremented through every 15 minutes. When two devices are close enough they exchange the ID currently in use. User A stores user B's ID and B stores A's in a "contact list". Then if A tests positive they authenticate this information with the health authority and upload their list of IDs from their infectious period to the server. Every user device periodically fetches this list from the server. Then the device compares these IDs to those in its own contact list and determines if the user is at risk. If a match is found and the system determines the user could have been exposed to the virus it notifies the user. In this case user B's device would perform these steps to find a match and a notification would appear in the app.

What makes the system "decentralized" is that the processing of contacts and risk levels is performed on the user's device. This is not the case in the "centralized" system.

Decentralized Privacy Preserving Proximity Tracing (DP-3T)

Decentralized Privacy-Preserving Proximity Tracing (DP-3T) was created by an international consortium of technologists, legal experts, engineers, and epidemiologists. In the white paper, the creators note that the design of the Google Apple Framework is similar to a specific case of DP-3T. Contact tracing apps based on DP-3T have been released in Austria, Belgium, Croatia, Germany, Ireland, Italy, the Netherlands, Portugal, and Switzerland. [162]

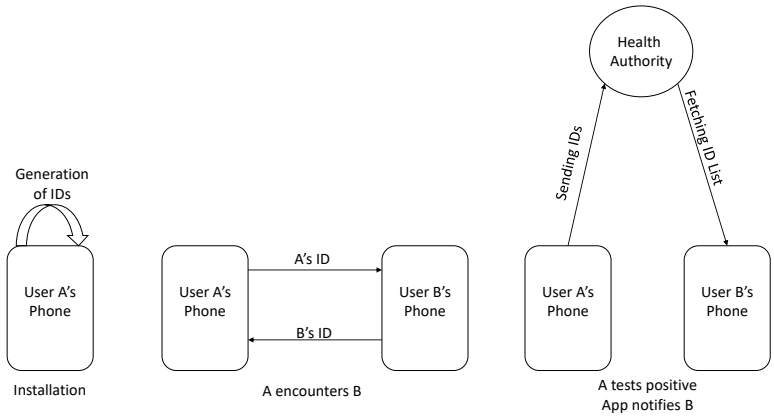


Figure 6.2: Visualization of the basic design of the decentralized system [137]

The DP-3T white paper actually outlines three similar protocols that are generally similar to provide developers a choice regarding the trade-off between privacy and computational cost. The three protocols are the low-cost design, the unlinkable design, and the hybrid design. The DP-3T white paper alone does not cover the secure mechanisms used to validate an upload of information, though the team did develop three protocols laid out in a different document. [162]

In all three DP-3T protocols, apps locally generate frequently-changing ephemeral identifiers (EphIDs) and broadcast them via Bluetooth Low Energy (BLE). Apps observe the beacons and store them together with a time and measurement used to estimate exposure risk, such as signal strength. The tracing process is supported by the back-end server that distributes a list of IDs for the apps to process to all the apps. The back-end acts only as a communication platform and performs no processing. If patients are diagnosed with COVID-19, they will be authorized by health authorities to publish a protocol-specific representation of their EphIDs for their contagious period. Apps periodically query the back-end for information and reconstruct the corresponding EphIDs of COVID-19 positive users locally. If the app has recorded beacons corresponding to any of the reported EphIDs, then the user might have been exposed to the virus. The app uses the exposure measurements of the matched beacons to estimate the exposure of the user. [162]

The low-cost protocol will be detailed first. The initial seed generation occurs once the app is downloaded. The app generates a random initial daily seed SK_t, for the current day t. Every day after the initial day the app rotates its secret key using a cryptographic hash function $SK_t = H(SK_{t-1}$. The EphIDs are generated from the SK_t. EphIDs should be regularly

changed and only broadcast for a single epoch, the length of an epoch will be L and is a system parameter. At the beginning of every day, the app generates a list of $n = (24 * 60)/L$ new EphIDs to broadcast for the day. A pseudo-random function such as HMAC-SHA256 is used with the SK_t and a public string as the inputs. This is then passed to a pseudo-random generator such as AES in counter mode and every 16-bytes is an EphID. This is displayed in equation 6.7. The app then randomizes the order in which the EphIDs are used and broadcast them for L length of time. [162]

$$EphID_1||...||EphID_n = PRG(PRF(SK_t, \text{“}broadcastkey\text{”})) \tag{6.7}$$

Where PRG is a pseudo-random generator, PRF is a pseudo-random function, and the broadcast key is some published string

When the app receives a beacon it stores the EphID, an exposure measurement like signal strength, and the day the beacon was received. For efficient storage, it is suggested that the entries be grouped by EphID. The app also stores the SK_t seeds that it generated for the last 14 days. The 14 days is a system parameter that defines the maximum period for which any data is stored. [162]

Once authorized as a confirmed COVID-19 positive user the user can instruct their phone to send to the back-end server the seed SK_t corresponding to the first day in which the user was considered contagious. The contagious window can be determined by a health authority or the user. After reporting the seed SK_t the app deletes SK_t and selects a new random seed to generate EphIDs from. [162]

The back-end collects the pairs of (SK_t, t) from positive users and stores them. Apps periodically download these pairs from the server. An app then uses the EphID generation method to reconstruct the list of EphIDs of the positive users and compares these to the app's contact list. The app also checks that the EphID was collected prior to when the SK_t was published to prevent malicious replay attacks. For a matching entry, the receive time and exposure measurement that was stored is assessed to compute the risk of exposure. [162]

The unlinkable decentralized proximity tracing protocol differs in that it does not disseminate a list containing the seeds of users who have reported to be COVID-19 positive. Instead, the EphIDs of positively diagnosed users are hashed and stored in a Cuckoo filter, which is then distributed to other users. [162]

For the unlinkable system apps generate an EphID by drawing a random 32-byte seed per epoch that is input to a cryptographic hash function. The output of the hash function then

has the leftmost 128-bits of the most significant 128 bits is taken as the EphID. The EphID generation is shown in equation 6.8 The app then stores the seeds used for every epoch in the last 14 days. [162]

$$EphID_i = MSB128(H(seed_i)) \tag{6.8}$$

Where $seed_i$ is a random value generated for epoch i, H is a cryptographic hash function, and $MSB128$ takes the leftmost 128 bits.

When an app observes a broadcast beacon it stores the EphID and epoch the beacon was received during as a hashed string. The hashed string is stored along with the exposure measurement of signal strength and the day that the beacon was received. It is recommended that the beacons are stored grouped by the hashed string. [162]

Once authorized as a confirmed COVID-19 positive user the user can instruct their app to send to the back-end server a representation of the EphIDs used during the contagious window. The user can select specific epochs that they do not wish to be included before uploading the seeds. [162]

The back-end periodically creates a new Cuckoo filter F and for each pair of epoch and seed received it inserts the most significant 128 bits of the hashed seed concatenated with the epoch and passed through another hashing function. This is illustrated in equation 6.9. Thus it inserts into the Cuckoo filter the hashed EphID and epoch pair. Then the back-end publishes the filter. Apps download the published filter F and check if any of its stored hashes are included in filter F. As before they check that the stored hash was collected prior to the publishing of the filter and use the stored exposure measurements to determine the risk of exposure. [162]

$$H(MSB128(H(seed_i))|i) \tag{6.9}$$

Where H is a cryptographic hash function, MSB128 takes the 128 most significant bits, and i is the epoch

The protocol notes that Cuckoo filters have a low non-zero probability for false positives. Thus making it possible for the system to report that it contains an element that was not within the input set. The parameters of the Cuckoo filter have been selected to make false positives highly unlikely even with heavy usage of the system over years. They calculate the filter will produce one false positive in a million users over a period of 5 years. [162]

The final hybrid design combines aspects of the two previous protocols. In this design, an app generates a random seed that is used to generate the EphIDs using a similar method to the low-cost design but only for a window of time. Thus if the window is two hours all of the epochs within those two hours use an EphID generated from the same random seed. At the beginning of every time window, the app picks a new random 16-byte $seed_w$. Then computes the EphIDs for that window by inputting the $seed_w$ and a fixed public string into a pseudo-random function such as HMAC-SHA256. The output of the pseudo-random function is then used as the input to a pseudo-random generator such as AES-GCM. The output of the pseudo-random generator is split into 16-byte chunks to obtain the EphIDs for that window. The generator is displayed in equation 6.10. An app broadcasts the EphIDs in random order. [162]

$$EphID_{w,1}||...||EphID_{w,n} = PRG(PRF(seed_w, \text{“}publicstring\text{”})) \tag{6.10}$$

When an app collects a Bluetooth beacon it will store the EphID, exposure measurement, and the time window that the EphID was received within. As before storage purposes suggest storing the beacons grouped by EphID. [162]

If authorized as a confirmed COVID-19 positive user the user can instruct the app to upload the relevant seeds for the windows within the contagious period. For efficiency, if the app does not have any entries in the contact list within a window within the contagious period it will not upload the seed for that window. The user also has the ability to select windows to not upload the seeds from. [162]

The back-end receives the seed and corresponding window from a positive user and then publishes them. Apps periodically download these pairs. The app then reconstructs the list of EphIDs of the positive users. The app then checks if it received any of the generated EphIDs during their corresponding window in the past. Then if there is a match of EphIDs the app calculates the risk assessment based on the exposure measurement and if the risk passes the threshold notifies the user. [162]

Google Apple Exposure Notification (GAEN) System

The Google/Apple Exposure Notification (GAEN) system is a framework and protocol specification developed jointly by Apple Inc. and Google to facilitate digital contact tracing. It was created to be used as an opt-in feature within COVID-19 contact tracing apps developed

and published by authorized health authorities. Originally unveiled on April 10, 2020, it is compatible with devices supporting Bluetooth Low Energy and running Android 6.0 "Marshmallow" or newer with Google mobile services, or iOS 13.5 or newer on apple devices. It is not compatible with Huawei devices released since May 2019 due to the US trade ban on Huawei. Exposure Notification apps may only be released by public health authorities. To discourage fragmentation, each country will typically be restricted to one app, although Apple and Google stated that they would accommodate regional approaches if a country elects to do so. As of May 2020 22 countries had received access to the protocol, these include Canada, Germany, Ireland, Japan, The Netherlands, Poland, Switzerland, The United Kingdom, and The United States [16]. [67]

The GAEN system is noted to be similar to the DP-3T hybrid design. The GAEN system uses one seed to generate the ephemeral identifiers of that day, and thus corresponds to the specific case where windows are 1 day long [162]. However in version 1.0 of the protocol did use a persistent tracing key for the generation of the identifier keys, this was altered in Version 1.1 [61]. As version 1.2 is the most recent version at the time of this document's creation GAEN version 1.2 will be the one detailed. It should be noted the difference between 1.2 and 1.1 is only in terminology. In the GAEN system terminology exposure notification is used in place of contact tracing [63].

In version 1.2 to log encounters between devices the system exchanges messages with nearby devices running the protocol. The encounter messages contain unique identifiers called Rolling Proximity Identifiers ($RPID$) and Associated Encrypted Metadata. The $RPIDs$ change every 15–20 minutes at the same time as the Bluetooth MAC address. The simultaneous change is done to help prevent third-parties from tracking a user. [62]

Every day a new random 16-byte Temporary Exposure Key (tek_i) is created using a cryptographic random number generator. From this tek_i two 128-bit keys are calculated, the Rolling Proximity Identifier Key ($RPIK_i$) and the Associated Encrypted Metadata Key ($AEMK_i$). $RPIK_i$ is created with the algorithm of equation 6.11. The generation of $RPIK_i$ requires an HMAC-based Extract-and-Expand Key Derivation Function (HKDF) they use the SHA-256 hashing function. The HKDF requires a Key, salt, info, and output length, respectively tek_1, no salt, a byte array that corresponds with the "EN-RPIK" string in UTF8, and 16 noting the 16-byte

output. $AEMK_i$ is created with the algorithm of equation 6.12. The generation of $AEMK_i$ is the same as $RPIK_i$ except the byte array corresponds with the "EN-AEMK" string in UTF8.

$$RPIK_i = HKDF(tek_i, NULL, UTF8(\text{“}EN - RPIK\text{”}), 16) \tag{6.11}$$

$$AEMK_i = HKDF(tek_i, NULL, UTF8(\text{“}EN - AEMK\text{”}), 16) \tag{6.12}$$

In the BLE specification, the MAC address is changed every 15-20 minutes to avoid devices being traced based on their MAC address. In GAEN every time the MAC address is changed a new temporary $RPID$ is generated. The $RPID_j$ is generated with the algorithm of equation 6.13. Where j is the Unix Epoch Time at the moment the roll occurs. The $RPID_j$ generation uses the $AES\,128(Key, Data)$ encryption function. The data portion is made up of a 6-byte array corresponding with the EN-RPI line encoded in UTF8, six zero bytes for padding, and Ti a 4-byte number of a 10-minute temporary interval, calculated as $unix_timestampdiv(60 * 10)$ where div stands for integer division. [63]

$$RPID_j = AES\,128(RPIK_i, UTF8(\text{“}EN - RPI\text{”})|0x000000000000|Ti) \tag{6.13}$$

Next, the 4 bytes of Associated Encrypted Metadata (AEM) are encrypted. This is done using the $AES\,128 - CTR(Key, IV, Data)$ encryption function. The $AEMK_i$ is used for the key, the $RPID$ as the initialization vector, and then the metadata is the input data. [63]

$$AEM_j = AES\,128 - CTR(AEMK_i, RPID_j, Metadata) \tag{6.14}$$

The entire BLE advertising payload is broken into three sections in order: Flags, Complete 16-bit Service UUID, and Service Data. The flags section contains the BLE general discoverable mode and shall have bit 1 set to 1, the UUID section contains the UUID of the exposure notification service which is 0xFD6F. The service data section contains two sections of payload. The rolling proximity identifier is the first 16 bytes, then the 4 bytes of associated encrypted metadata. Contained within the associated encrypted metadata is byte 0, of which bits 7:4 are the system version, bits 3:0 are reserved for future use. Within byte 1 is the measured radiated transmit power of Bluetooth Advertisement packets, and is used to improve distance approximation. Bytes 2 and 3 are reserved for future use. [62]

Once a registered health authority has confirmed the infection of a user, the user's Temporary Exposure Keys tek_i and their respective interval numbers i for the past 14 days are uploaded to the central reporting server. Apps then download the list of pairs and individually regenerate every Rolling Proximity Identifier. The $RPIDs$ are compared against the app's local encounter log. A single encounter is also only stored for 14 days in the encounter log of an app. If a matching entry is found, then the app decrypts the associated metadata for that encounter and performs a risk assessment based on the transmitted power level. If the risk level passes a developer determined threshold the app presents a notification to the user warning them of potential infection. The method through which daily encryption keys are transmitted to the central server and broadcast is left to be defined by individual app developers. [67]

It is noted in the specification that the associated encrypted metadata does not get decrypted unless a match occurs. This data then needs to be sanitized and validated as the associated encrypted metadata is not authenticated. It also notes that it is computationally infeasible for an attacker to find a collision on a Rolling Proximity Identifier. [63]

6.2 GPS Based Contact Tracing Systems

GPS contact tracing methods focus both on who someone was near as well as where they went. GPS data is recorded, and then if someone tests positive that information is used in a variety of possible ways to inform other users of their exposure risk. A few countries have implemented GPS based tracking of their citizens, notably Iceland, Colombia, India, Israel, Jordan, Kuwait, Norway, Qatar, and Russia. It should be noted the Norway Smittestopp app was taken down. Some of these countries have released an open-source codebase for their implementation. As with the Bluetooth system, there are different systems that can use GPS for contact tracing. The main ideas being centralized and decentralized models. In both systems, location data is being sent to a server. The difference is in the centralized version the comparisons to other users happen on the server-side. While in the decentralized version the comparisons happen on a user's device.

There have been different ways in which health authorities have used the GPS information of patients with COVID-19. In some countries, there are websites that are updated with locations and times that a patient was in that location. Then users can check the map regularly and see if they may be at risk [104]. In South Korea updates from the government that detailed the places that patients with COVID-19 visited were pushed to citizens' phones. Some citizens

logged this information onto a map themselves and in some cases were even able to re-identify the sick patient from the information that was provided [104].

Published maps can solve the problem that many of the other digital forms of contact tracing have. That being not all members of the populace have devices such as smart phones that can perform digital contact tracing. The published map system allows for people that do not use the app or have a compatible device to go online and see if they may have been exposed.

6.2.1 Centralized GPS Data Contact Tracing

There are two types of systems that are centralized GPS contact tracing. In what we will call the Devie Held GPS Data method. The data is stored on the device until a user uploads it to the server. In the Server Held GPS Data Contact Tracing the data is continually or regularly uploaded to the server once the user installs the app.

Device Held GPS Data Contact Tracing

The Icelandic government released an app called Rakning C-19 which uses GPS data in a centralized way. Though there is no specific white paper for the protocol they did release an open-source code base that has a basic description of how the system operates.

After downloading the app users register with their phone number. The app sends the user's phone number, their locale, and push notification token to the server. The app requests permission to track location in the background. When permission is granted the app stores geolocation updates on the device in an SQLite database. If the Contact Tracing Team needs a user's assistance in tracing the contagion, they will send a request to the back-end. When the back-end receives a request for data, it marks the user for data collection and triggers a push notification in the app. The next time the user opens the app, it checks if there's a data request and asks the user to approve the request before sending the last 14 days of geolocation data to the back-end. This information is then used by the contact tracing team to alert citizens of possible places and times of exposure. [3]

Server Held GPS Data Contact Tracing

The apps released in Russia, Kuwait, and Qatar all do not have a white paper or open-source code base to inform on how their GPS based contact tracing system work. It is known that the systems are actively uploading information to a server however through what information

has been released or analysis of the app's communications. For example, the Russian app Contact Tracer logs GPS location data and can inform users of how many infected individuals are nearby in real-time [126]. It can be inferred that their implementation regularly connects to the server and uploads the current GPS details of a user. Then in the case of a positive diagnosis, a user can register that information to the system and in the back-end, it will send alerts to any users who were in the same locations at the same time as the diagnosed user.

6.2.2 Decentralized GPS Data Contact Tracing

The Israeli government released an app called HaMagen that uses GPS data in a decentralized way. Though there is no specific white paper for the protocol they did release an open-source code base that has a basic description of how the system operates.

In the HaMagen system the app stores the user's GPS data on the device. When a patient tests positive for COVID-19 and they consent to the upload of their GPS information, their location data is passed to the back-end server. The back-end server collects this information into a single list that the apps can fetch. An individual app then pulls the list compares its own stored history to the list. If there is a match of location and time the app notifies the user. The app only stores the last 14 days of location history, and this GPS system is paired with a Bluetooth token exchange system. Thus, if a person is diagnosed with COVID-19 both the Bluetooth and GPS data from the last 14 days is sent to the server. They note that only possible points of exposure are taken into account, likely determining this through a tracing interview. [106]

A difference between this version of decentralization and the Bluetooth system's decentralization is that in this case, the authority is receiving the full location history of a patient. Where in the Bluetooth system all that the authority receives is a list of keys or IDs that the server publishes. Section 4.4 discussed how this information can be used to determine things like someone's home, place of work, habits, homes of friends and family, and other private information. It is a very detailed collection of information about a person's life.

Chapter 7

Methodology of Assessing Contact Tracing Applications

When the idea of digital contact tracing was introduced, privacy advocates got to work considering the privacy implications. The technology was moving quickly, as the tech industry wanted to help flatten the curve, and suggested systems needed to be analysed and considered before being implemented. This is an important step, the tech industry in the last 20 years has been famous for "moving fast and breaking things". It is their motto for finding problems early so that they can be solved. However, when it comes to privacy it is difficult to impossible to fix what has been broken. Once someone's social insurance number is out in the world their identity can be stolen for years to come making life annoying at best and at worst very difficult [5]. With the world panicking over a pandemic hate crimes against marginalized or racialized communities was on the rise [7]. If the system leaked or gave too much information about a patient out the effect on someone's life could be grave [7]. The South Korean system of notifications led to patients being identified and harassed online [171].

It was also clear immediately that the contact tracing apps were an opportunity for governments or corporations to collect a lot of information quickly and easily. An e-pass that citizens are required to have to use public transit or go to the grocery store is a control of many citizen's movements. An app that collects all the GPS data on a device and uploads it to a server is detailed information about an entire populace of individuals that a government might not have otherwise been able to get. This information is powerful. Even without a name, it is easy to determine who someone is, where they work, who they are close with, their daily habits, and create predictive models of where they will be in the future, all detailed in section 3.2. It could

give a government a lot of power over its citizens, its critics, and its political opponents. In the UK more than 150 scientists and researchers released a joint statement about their concerns over the release of a contact tracing application [14].

To fully consider a contact tracing app it needs to be reviewed on two fronts. The privacy of the system and the security of the system. The privacy of the system is about looking at what the developers and controllers intend to do with the app. What data they intend to collect, how they intend to use that data, who has access to the data, etc. The kinds of protections the governor of the system has on the data are also important to this. The security of the system is about looking at what someone from outside of the system could do to or with it. These are attacks against the system or unintended uses of the system. Thus the two considerations are what the organization is saying they want to do and another is what someone outside of the organization could do.

This thesis proposes a method to compare the privacy and security of differing contact tracing apps. The privacy comparison will be based on how well an app meets a set of principles laid out by experts in the field of security and privacy. The security comparison will be based on how severe the most severe vulnerability to the system is. 55 apps from a variety of countries will be researched. Of those 5 representing different methods of contact tracing will be selected. These representatives will be used to create an assessment system that can then be applied to other applications.

7.1 The Basis of Our Methodology

The most widely used method to assess and compare the severity of a computer system vulnerability is the Common Vulnerability Scoring System (CVSS). The CVSS assigns severity scores to vulnerabilities, allowing those responsible to prioritize responses and resources according to the threat. Scores are calculated based on a formula that depends on several metrics that approximate the ease and impact of the exploit. Scores range from 0 to 10, with 10 being the most severe. The equations used by the CVSS were developed by the CVSS Special Interest Group (SIG) who framed a lookup table by assigning metric values and a severity group (low, medium, high, critical) to real vulnerabilities. Having defined the acceptable numeric ranges for each severity level, the SIG then collaborated with Deloitte & Touche LLP to adjust formula parameters to align the metric combinations to the SIG's proposed severity ratings.

Thus the equations were created by qualitative rankings of known vulnerabilities then working backwards to create a quantitative method that could be applied to future vulnerabilities [47].

The method of comparing contact tracing apps will be created similarly to the CVSS. First, the apps will be researched. It will then be determined whether they do or do not meet certain privacy principles. Then 5 representatives will be selected. The apps will be assigned, based on their privacy, a grouping of green, yellow, or red, corresponding to good, fair, or low privacy respectively. Then a rating system will be determined and the assigned groups of the 5 apps will be used to determine the thresholds of the group. It was decided that the rating will be based on how many of the principles were met by an app.

Then the security of the 5 apps will be determined. A series of potential vulnerabilities will be theoretically applied to the systems and their ability to prevent or mitigate the attack will determine a score based on a predetermined rubric. The score, in this case, is higher the more severe that vulnerability is to the system. Then the apps will be assigned, based on their security, a grouping of green, yellow, or red, corresponding to good, fair, or low security respectively. These groupings will then be used to determine how to apply the scores from the rubric to create a method of ranking other applications.

The CVSS was not directly used for the scoring of the vulnerabilities in this thesis. This is because the CVSS is designed as a way to compare vulnerabilities of systems not the security of a system as a whole. Also, a system that is more specialized to the unique requirements of contact tracing could provide more actionable information. The CVSS does not consider privacy at all. For the rubric of the security assessment, some metrics are similar to those of the CVSS, attack complexity, required privileges, scope, confidentiality, and the division of the rubric into exploitability metrics and impact metrics is taken and adjusted for contact tracing from the CVSS.

7.2 Privacy Principles of Contact Tracing

If you use software, then you have likely accepted the terms of a privacy policy. This is often a statement about what data the software collects, how it is used, and who is responsible for it. Some countries have laws requiring that a company disclose specific information to users about the data collected from the user, as well as how that data must be protected and for what purpose that data can be collected. There are also ways in which applications specifically are prevented from using or gaining access to certain pieces of data without user consent being

given. These are the app permissions that users have to click okay on before the app can use a feature.

All of these rules, regulations, guidelines, consent requests, etc. are based on privacy principles. A set of fundamental ideas and beliefs about the level of privacy that users should have and the control they should have over the data that is about them. When new technology or ideas are created it is important to return to those privacy principles and create a set of principles that are designed to specifically guide the new creation so as to not erode the privacy of citizens.

Getting ahead of the technology is difficult, as it moves quickly, but the best way to design the system is to start with principles. In April of 2020, the Office of the Privacy Commissioner of Canada released a Framework for the Government of Canada to Assess Privacy-Impactful Initiatives in Response to COVID-19 [128]. The government's framework identifies nine guidelines that should be followed by any solution intending to use data to assist in ending the health crisis. These are legal authority, that is all organizations must continue to operate under an identified lawful authority when collecting, using, and disclosing personal information. Necessity and proportionality, ensure that the measures the institution wants to take are both necessary and proportional both to the situation and to the amount the measures will affect the situation. Purpose limitation, the information that is being collected to alleviate the effects of COVID-19 must not be used for anything else. De-identification and safeguarding measures, where every possible personal information should be de-identified, or aggregate data should be used. Vulnerable population considerations, there are often unique impacts of measures on vulnerable groups. Openness and transparency, clear and detailed information about measures being taken now and in the future should be available to the people. Open data, any data made public should be considered carefully to weigh the benefits and risks before the release of the dataset. Oversight and accountability, there need to be specific provisions for what authority is overseeing the measures and accountable for them. Time limitation, any privacy-invasive measure should be time-limited and end when they are no longer required to alleviate the health crisis. [128]

The American Civil Liberties Union created a list of basic principles for evaluating technology assisted contact tracing methods. Their 14 principles focus on the system being voluntary for citizens to use, that as little data as possible is collected, that the system is secure, that the system is designed to be effective and useful so that the risk is worth the reward, and that there is an end to the systems use. Their principles are not displacing non-technical

measures, Voluntary, Non-punitive, Built with public health professionals, Privacy-preserving, Non-discriminatory, Minimal reliance on central authorities, Data minimization everywhere, No data leakage, Measurable impact, Have an exit strategy, Narrowly-tailored to target a specific epidemic, Auditable and fixable, Sustainably maintained. [57]

The Chaos Computer Club, Europe's largest association of hackers that has been providing information about technical and societal issues, such as surveillance, privacy, freedom of information, hacktivism, data security for more than thirty years, also released a set of ten requirements for the evaluation of contact tracing apps. These include the societal requirements, that the system has epidemiological sense and purpose, that it be voluntary and free from discrimination, that privacy is fundamental to the system, and that it is transparent and verifiable. Then there are the technical requirements, that the system has no central entity users are required to trust, that data collection is minimal, that data collected is anonymous, that there is no creation of centralized movement or contact profiles of users, that any identifiers are unlinkable to the user, and that the communications of the system be unobservable. [27]

There were also a set of privacy principles released by the University of Waterloo Cybersecurity and Privacy Institute and signed by security and privacy researchers from across Canada. This set of privacy principles will be the main focus of the privacy review of contact tracing apps performed in this thesis.

7.2.1 Waterloo Privacy Principles of Contact Tracing

In May of 2020, the Waterloo Cybersecurity and Privacy Institute released a Coronavirus statement signed by security and privacy researchers representing twenty universities across Canada. The signatories claim that the development and deployment of contact tracing apps in Canada is being done without sufficient technical independent expert review. They argue that the ten principles that they lay out should be applied to the development and deployment of any tracing app. These ten principles are independent expert review, simple design, minimal functionality, data minimization, trusted data governance, cybersecurity, minimum data retention, protection of derived data and meta-data, proper disclosure and consent, and provision to sunset. The meaning of these principles will be detailed. [78]

Independent expert review refers to the design and implementation of the app being subject to open reviews by software security and privacy experts who are not connected with the development team or organization. This review should be performed in advance of deployment,

not post-deployment. Such a review ensures that the objectives of privacy protection have been properly implemented within the design when the first users are downloading it. [78]

The simple design principle is that the app should be developed using the simplest approach possible to perform the intended function. The simpler the design the faster it can not only be implemented but reviewed. The more complex the system is the longer it will take reviewers to ensure it conforms to the design specification and pertinent principles of data protection. This includes any supporting servers and their code as well [78]. It also has the additional effect of being easier for the public to understand the system, and it is important that the public trust any system that is released. If the system is not trusted it will not be used. A system that is not used by enough people to be effective is not only a wasted effort but an unnecessary risk of privacy to those who are using it. No matter how low the privacy risk may be.

Minimal functionality is referring to the app only providing the necessary functionality to allow for contact tracing. There should be no additional code or secondary purpose incorporated into the app. Any additional functionality must be part of the review performed and included in any design documentation. A consequence of secondary functionality will be a longer public debate about the value of deploying the app. [78]

Data minimization is common among privacy principles. The only data collected by an application should be what is required for contact tracing purposes. Whenever possible, contacts (and any other required app data) should be retained on the device where they are collected and should only be shared with other users or to a central repository if required for a contact tracing incident. Any data collected and stored creates an obligation to properly manage this data, complicating the design of the app. [78]

Trusted data governance is about who controls the data, who has access to the data, how the data can be used, overall who is accountable for the data. If data is transmitted to a central repository, the repository should be a trustworthy actor subject to public oversight. A government or health sector agency subject to investigation by privacy commissioners/ombudspersons should be used to store contact data. No private-sector data repositories should be allowed access to contact data. A properly managed central repository can enforce data protection and cybersecurity of sensitive data. Trust in the authority over the central repository is trust in the security of the system. [78]

The cybersecurity principle is about the security of the entire system. A tracing app is part of Canada's critical infrastructure with the potential to send many people into isolation or quarantine. Thus, the highest level of cybersecurity must be implemented for all aspects of the contact tracing app, including the collection of the data on the device, the tracing app itself, the communication channels used to move data, and any central locations. All conceived malicious attacks should be considered. The security of the system includes audits and monitoring to ensure breaches do not occur or can be contained if they do happen. [78]

Minimum data retention , this principle is about the length of time that the data collected can be held. The data collected should only be retained for the lifetime of its intended purpose. For COVID-19, data should only be retained for the infectious period for the person carrying the device and any data stored centrally (or with other devices) should be permanently deleted after it has been used for the contact tracing required. [78]

Protection of derived data and meta-data , derived data is data that might be created while processing other data or is information that can be inferred from the data available, meta-data is the information about app use for example. Derived data and meta-data allows for sensitive inferences about users. Derived data should only be used with consent and should be protected by mechanisms to prevent re-identification. No meta-data should be collected, stored, or used in the analysis of contact traces. This also relates to the data minimization principle because collecting or creating data that is not required is not minimizing data. [78]

The proper disclosure and consent of the user is important for the trust of the system. The user must be made aware, clearly and understandably, what data is collected about them and how it is used. Uninformed consent is not truly consent. If a user does not have the opportunity to understand what they are being asked to consent to they cannot consent to it. Thus, information must be made available to all potential users. This information needs to be written in language they understand and detail entirely the data that is being collected and its uses. If there are uses for this data beyond the contact tracing functionality, these must be made explicit and separate consent received for each such use. This disclosure and consent should be renewed regularly to ensure both the ongoing need for the tracing functionality and the users' commitment to continuing to participate. The disclosure also needs to be easy to find so that users may review it periodically. [78]

A provision to sunset is a requirement because this health crisis should not be used as an opportunity to enact something with a lifetime longer than the crisis. There should be provisions to sunset the app and delete its collected data after the COVID-19 crisis is contained. Data collection should be automatically terminated and notification of all participants should occur. Any residual data should be deleted as soon as the app is no longer used. [78]

7.2.2 Methodology of the Contact Tracing Privacy Review

The privacy review began with searching for all of the contact tracing apps deployed or in development around the world. There were a variety of lists available online of contact tracing apps either deployed or in development. For the ease of access, the list available on Wikipedia [168] was used as a jumping-off point to find the government websites and published information about the specific apps, as well a list published by XDA news of countries using Google and Apples Exposure Notification system [136]. To find the information on the app official government websites as well as the google play store and iOS app store were searched for privacy policies or statements, FAQs, open-source code, white papers, and any other information available about the app. A focus was made to answer the questions posed by the principles of the Waterloo statement. News about the app was also searched to determine what external sources had learned about the app.

As the review was not limited in scope and intended to review apps from many countries it should be noted that in the case of English language material not being published the translation was performed by the google translate tool. This is a limitation due to the reviewer only being fully literate in English. Every effort was made despite this to ensure that the review was thorough.

The apps included in the review are any discovered whose function was to notify the individuals that a COVID-19 patient came into contact with during their infection period. Apps that are of similar or overlapping function but do not have a contact tracing function within them are not included. There were also apps discovered for which the information on their operation was lacking to the degree that it appeared meaningless to include them within the review directly. Though a principle of the privacy review is disclosure. Thus these apps will be discussed but not directly compared with the others.

Once the information was collected on the apps they were compared to the ten privacy principles. It was determined that the app could either meet the principle, partially meet the requirement, or not meet the requirement. Partially meeting the requirement typically consists

of having the requirement but not to the extent of the strict interpretation of the principle. An app that had no information directly about a principle and thus it cannot be said whether it met or did not meet the principle is treated as not meeting the principle. This allows the review to follow the principle of disclosure. If the user cannot find the information about the principle then they cannot make informed consent. The entire system of contact tracing relies on trust. If the public cannot learn the information about the system the public has to assume that the principle is not being met. Transparency is trust.

Criteria for the Privacy Principles

Independent Expert Review is met if the app was reviewed by independent experts who provided a documented review of the app prior to the release of it. Half met means that there is the potential that it can be reviewed by independent experts, something like the source code being available. Not met means that there was no clear review process performed or able to be performed.

Releasing the source code does not count as meeting the requirement because it is clear that the idea of the principle is that the review is performed prior to the release of the app. In most cases source code is released simultaneously or post the release. As independence is not defined in the principles the criteria that to be independent you must not be a direct part of the organization that is releasing the app. This means that while a review by a part of the government containing experts in the field of security is appreciated and makes the principle half met, it does not completely meet the criteria.

Simple Design is met if the app is based on a publicly available white paper or equivalent that details the protocol and design being used. Half met means that there is information available online about how the system operates though this may be informal or piecemeal across released information. Not met means that there is no documented or reviewable basis for the design of the system.

Minimal Functionality is met when the app only does what is required for tracking based on the protocol or design. Half met means that it has perhaps one or two additional functions that are detailed and do not require more information that the system already has for tracking. Not met means that the system has a lot of extra functions.

There is the issue that some have not released their protocol, for comparison to the system function. If this is is the case the principle is based on what the system needs to do for contact tracing using the method that it claims to use. This can require the reviewer to compare similar methods that have released more information. Also, even a function still closely related to contact tracing is an extra functionality. For example, symptom tracking, though relevant, is an extra function. The only function allowed is providing information to users from government releases about best practices for safety and the current status of the pandemic in the country. This is because almost all of the apps have this and it does not require more data from the user.

Data Minimization is met if the system does not require the user to enter any personal information. Half met is if the system requires one or two pieces of information that are not typically considered an issue. For example, a phone number though it can be used to identify someone is not considered a privacy violation. Not met means that the system requires more information than the contact tracing should require. Things like detailed health information, a national ID, detailed GPS data. All personal information.

The requirement to meet this principle may seem strict, however, the contact tracing protocols of ROBERT, DP-3T, and GAEN do not require any information from the user. This is because a notification to the user can be passed through the app directly. What is allowed is if the app asks for the date a user took a test, the day symptoms started, and the illness status of the user. Since this is typical information required for determining the contagious window of the user. Information that is tied to an identity or that is identifiable and superfluous to tracing is more interesting to us.

Trusted Data Governance is met if the data is owned by the government Department of Health or an equivalent, does not leave the country, will not be accessed by any other body of government or third-party, and the authority is subject to public oversight. Half met could be that the owner is known but the other assurances are not addressed. No means that the owner is either not known or is a third-party, it can be accessed by other groups or some other clear violation of the principles.

Cybersecurity is met if there are servers in the country secured to industry standard, the data is encrypted in transit and when stored everywhere in the system, and the system is monitored and audited to prevent breaches. Half met means that there are implied or basic statements

towards the security of the system but no detailed information on what is being done. Not meeting the requirement means that the security of the system does not meet standard practice.

Minimum Data Retention is met when data is held only for the infectious period of COVID-19 as stated by the WHO to be 14 days [167]. Half met means that the data is being held for longer than 14 days, but a stated length of less than a year. Not meeting the requirement means that the data is held for an undetermined length of time or over years.

Protection of Derived Data and Meta-Data is met if there is a statement to the effect that no data of this kind is created, any data of this sort is deleted, or that it is secured to an industry standard. Half met means that while not specifically mentioning these kinds of data there is some information about what is done with it. For instance, an app may aggregate app usage metrics and store them securely for a period of time, or delete IP addresses from server logs. Not being met can mean a few things to the nature of not being mentioned or the information being sold to a third party.

Proper Disclosure and Consent is met when the use of the app is voluntary, all data being released from the app is with the consent of the user, the app can be deleted or turned off at the user's discretion. Also that the privacy policy is easy to find, detailed, and clear in language. Half met means that the app use is voluntary and there is a privacy policy however, the policy may not be completely detailed. Or privacy information was found but there was no clear privacy document. Not being met is if the app is mandatory for citizens to use, or there are no privacy statements that can be found.

Voluntary for this principle means not only that the government is not fining or criminally treating those who do not use the app but that there are no negative impacts on someone's life for not using the app, beyond those associated directly with contact tracing. For example, if the app is voluntary to download but citizens cannot enter a grocery store or public transit without showing that they have a green symbol on their app then it is not considered voluntary. Some countries have privacy policies, notices, or statements, these are all treated as a privacy policy for this review.

Provision to Sunset is met when it is stated clearly that the app and all its information will be deleted at some time, and there is a clear method for determining when that will be. Half met is when there is a statement to the effect of sunsetting the app but a definite time or method

for determining when is not given. Not met is when it is stated or implied that the app will continue to be used beyond the time of the health crisis.

In a few countries, they state they will sunset the app once the pandemic is over. A simple statement like that is considered as half meeting the requirement because there are no clear criteria for when the pandemic is over. It could be when a vaccine program is complete, or if they have had no new cases for a time, or no deaths for a time, etc. If there is no clear way that the end of the pandemic will be determined then this criteria has not been fully met.

7.3 Methodology of the Contact Tracing Application Vulnerability Analysis

As described earlier, 5 apps that are representative of the different contact tracing schemes will be analysed through their potential vulnerabilities. A series of potential vulnerabilities will be theoretically applied to the systems and their ability to prevent or mitigate the attack will determine a score based on a predetermined rubric. The score, in this case, is higher the more severe that vulnerability is to the system. Then the apps will be assigned, based on their security, a grouping of green, yellow, or red, corresponding to good, medium, or low security respectively. These rankings will then be used to determine how to apply the scores from the rubric to create a method of ranking other applications.

The review will make a few assumptions about the system that is being assessed. First, unless there is compelling evidence to suggest otherwise it will be assumed that the information the authority has provided about the app is accurate to how it was implemented. This review is not assessing whether there are vulnerabilities caused by human error in the implementation. Second, if it is not stated in the documentation found then it is not happening in the system. For example, if there is no information about the system using HTTPS when transporting data it will be assumed that they are not.

The actual implementation of a vulnerability into an exploit against the system is beyond the scope of this thesis. The systems were not attacked in any way to perform this assessment. The review looks only at the theoretical side of the protocol and system to determine if there is a vulnerability. It should be noted that this is the easiest part of the system for the developer to get right. As always, the theoretical world can be perfect, while the real world can only strive for perfection.

7.3.1 Contact Tracing Application Vulnerability Rubric

The rubric created for the assessment of vulnerability is displayed in table 7.1. This was inspired in ways by the CVSS, with those aspects adjusted to the specific issue of contact tracing. The areas similar to the CVSS are attack complexity, required privileges, scope, confidentiality, and the division of the rubric into exploitability metrics and impact metrics.

The rubric contains eight metrics, four focused on exploitability and four focused on impact. Each metric is given a score between 0 and 10. A 10 is the worst-case scenario for that metric, while a 0 is the best case. A vulnerability can score anywhere from 0 to 10 for that metric. Generally, the scores are 0, 1, 4, 7, 10, corresponding to ideal best case, practical best case, medium case, bad case, worst case. These values were chosen because in security there is often the noted difference between perfectly secure and practically secure. In this rubric, if something were impossible it would be scored 0, while something practically impossible would score a 1. Then the rest of the numbers are evenly distributed up to 10. The metrics will be explained and can be seen in the summary within and below table 7.1.

The exploitability metrics are Access, Knowledge, Complexity, and Effort.

Access refers to the privilege level in the system that would be required to exploit a vulnerability. The lower the access level required, the higher the score given. An attacker being able to perform an attack with minimal privilege on the system is worse than high-level privilege being required. This is because higher-level privileges are protected and more difficult to achieve. An attacker is also most likely to take the path of least resistance to achieve their goal. If it can be done with less effort then that will be the exploit they choose to use. The adage of "work smarter not harder" is also true for attackers.

Knowledge is about how experienced an attacker would have to be able to be to take advantage of a vulnerability. The more specific the knowledge required to build the implementation the lower the score. Correspondingly the less knowledge required to build the implementation the higher the score as a novice being able to build an exploit is the worst-case scenario. The less knowledge required the more likely the vulnerability is to be exploited because the pool of people capable of it is larger. As well even experts in the field still will go for the path of least resistance, so the attack that a novice could build is still one that an expert might use.

Complexity is about the technical requirements of a possible exploit. This is split into two, the technical complexity and the build complexity. Technology refers to computing power. An exploit that is very computationally intensive is less likely to be used as a vector of attack because fewer people will have those resources available to them to use. Thus an exploit that can be performed with a cell phone is ranked as a 10, while one that might require a server farm is considered a 1. The build complexity is about how many people would be required to create the exploit. If one person of the requisite knowledge level on their own build it within a week or would a whole team of people be required to build it within a month. The smaller the group the higher the score.

Effort is also split into two metrics, planning and human. Planning is about how much organization has to be done to implement the exploit. If there are a lot of different pieces to the attack, a lot of components working together, especially if they cannot be automated, then that is a lot of effort. The more effort that an exploit requires the less likely that it is to be used. Thus, an attack with minimal components that can all be automated will score higher, as that is the worst-case scenario. Then human effort is about how many people need to be involved to perform the attack. Having a lot of people involved complicates things even if they are not aware of their involvement such as a phishing attack. Also, the more people involved the harder it becomes for an attacker to not get caught. One person keeping a secret is easier than twenty.

The Impact metrics are Scope, Impact, Detection, and Damage.

Scope is about how many people the attack could affect. An attack that can target one specific person is dangerous, but if it can affect a whole section of the user base or the entire user base that is worse. Personal attacks should not be completely ignored as there are a lot of ways that a personal attack could be very harmful. However, it does remain true that everyone losing their privacy is worse than one person losing their privacy on a scaled system. Thus, if it could impact the entire user base it is given the worst-case score of 10.

Impact is also split into two. First, there is data. Data impact is either how dangerous the data that the attacker gains is, or how far into the system the attacker is able to place false data. An important thing is that an attacker might not just be trying to steal information, they could be trying to cause havoc for some reason. The attacker gaining information that is hard to use or tie to someone's identity is less dangerous than getting information that they can directly use.

If the attacker is trying to cause chaos or distrust in the system then the information getting deep into the network will be worse.

The second part of impact is trust. Trust is very important to contact tracing. These are government systems that are being put in place to try and save lives. To try and alleviate a health crisis. Since they are a government system they will reflect onto the government that implemented them. If a system were found to have a critical vulnerability that could be a powerful factor in the people's trust of their government. This tied together trust could go the other way and chip away at governmental trust if the system were to be vulnerable. As well even if the contact tracing system is sunset a few months from now it is possible that in the future an epidemic of another illness could breakout and a contact tracing system would be required again. When that happens the trust in the system now will impact how people view that future system.

Detection is how quickly an attack using this vulnerability could be discovered. If the system monitors for these kinds of attacks and alerts the caretaker to the danger even a major attack could be mitigated quite effectively. The sooner an attack is known the less damage that it can do. An attack that is never noticed is one that can be done many times and potentially lead to much greater damage than it would otherwise.

Damage is the final metric. This is split into two as well, the system and the user. The system damage is about how much would need to be done to get the system back to where it was prior to the attack. This is not about what would need to be done to prevent the attack from occurring again. If the attacker can break the system, can make it so that someone has to go in and fix things, route out false data, and clean house that is worse than an attack that leaves the system usable. The reasoning being that a hospital losing health data is bad, but the hospital's systems being unable to operate as well is even worse as seen in ransomware attacks [114]. As this is a healthcare system it can be viewed similarly. The damage to the user is about how long the user will feel the effect of this attack.

Once the scores have been determined for each of the rubric metrics an overall score for the vulnerability will be determined. Firstly a 0 for any of the exploitability metrics results in a 0 for the entire vulnerability. This represents that if the vulnerability was impossible to exploit for some reason then it is not truly a vulnerability. If none of the exploitability metrics are scored as 0 then an average of the scores is determined. First, the metrics split into

two are averaged, then all of the metrics are averaged together. The average was chosen as a simple way to compare the different vulnerabilities, other methods that could be used will be discussed in 9.2. The method for calculating the score is shown in equation 7.5. Where $A = Access$, $K = Knowledge$, $C = Complexity$, $T = Technology$, $B = Build$, $E = Effort$, $P = Planning$,$H = Human$, $S = Scope$, $I = Impact$, $De = Detection$, $Da = Damage$, and $\wedge =$ logical AND.

For many of the metrics, it was determined that the more work the vulnerability takes to exploit the less dangerous it is to the system. This idea appears in the scoring of all of the exploitability metrics. This was noted during the de-identification review of chapter 3. It was noted that if there were two avenues of attack and one required less work that is the attack that would be used. Work, in this case, is referring to any of the exploitability metrics. It could be less access or knowledge required, less complexity, or less effort. The easy attack was the more likely attack, even when experts in the field were performing the attacks.

$$Complexity = (Technology + Build)/2 \tag{7.1}$$

$$Effort = (Planning + Human)/2 \tag{7.2}$$

$$Impact = (Data + Trust)/2 \tag{7.3}$$

$$Damage = (System + User)/2 \tag{7.4}$$

$$Score = \begin{cases} 0 & if A \wedge K \wedge T \wedge B \wedge P \wedge H = 0 \\ \frac{A+K+C+E+S+I+De+Da}{8} & else \end{cases} \tag{7.5}$$

Also, when researching different systems, some had a bug bounty program set up. That is if someone could find a vulnerability or bug in the system, and privately disclosed the information, they would be rewarded with monetary compensation. However, on some of these systems like that of India, it was noted that they would not consider it a vulnerability if the device needs to be rooted to operate the exploit [119]. In this review, if the phone needs to be rooted this is still considered a valid attack. The access score will, however, be low. The reason for this is that there is a difference between root access on a remote server or computer that the attacker does not own or have physical access to and a device that they control. When an app is released and the attacker can download it onto a device that they completely control, root access is much easier for them to achieve. In this case, when an attacker owns the device that the system is on it to not consider what they can do with root access to the device is ignoring real

world implications. An attacker will root their device to perform an attack. It is not difficult for them to do. Thus, if an attacker has ownership and physical access to the device it should be considered possible that they have root access to it.

Table 7.1: Vulnerability Rubric

Score	0	1	4	7	10
Access	Prevented at any privilege level	High level of access required, need root access.	Medium level of access required, need higher access than regular user but not full root access	Some level of access required, need access beyond the user interfaces	Lowest level of access required, regular users could do this without higher privilege
Knowledge	Full knowledge of the system does not give someone the tools to attack it	Expert level of knowledge required, expert in the field with many years of experience in a specific subject	Advanced level of knowledge required, individual with career in the field	Intermediate level of knowledge required, graduate level student	Novice or fundamental level of knowledge, undergraduate level student
Complexity	Not possible with modern computing power	Highly complex and resource intensive, this requires a large amount of computing power	Medium complexity, requires high end computers	Low complexity, requires an average consumer computer and some tools need to be created	No complexity, this can be performed with minimal computing power, directly on a phone, with a single-board computer etc.
	Not reasonably possible for any group to design	Large group actor would be required	Multiple people could create with a month of time invested	Single person could create with a month of time invested	Single person could create in less than a month

Table 7.1: Vulnerability Rubric

Score	0	1	4	7	10
Effort	Extremely high effort, many components with very specific timing that cannot be automated	High effort lots of components that need to all work perfectly	Medium level of effort, several components required	Low effort, a couple of components required	Very low effort, minimal components or steps
	Would requires a large percentage of the user population to work together	Many people required to help (wittingly or unwittingly)	Multiple people required to help (wittingly or unwittingly)	One or two people working together	One person can perform alone
Scope	Does not affect anything in the system	Affects a single person, Attacker can gain information for one person that they may know	Affects a small group, attacker could target everyone they know	Affects a large group, Attacker could target an entire section of the user base or a demographic	Could affect everyone in the system
Impact	Attacker gains no access to alter or view the system	Attacker gains access to de-identified data that is not practically re-identifiable or introduces false data on the device	Attacker gains access to de-identified data that could with effort be re-identified or introduces false data on the server that will be deleted	Attacker gains access to data that with minimal effort can be re-identified or introduces false data that is persistent in the system	Attacker gains access to personal information and identities or introduces false data that is persistent and indistinguishable from real data

Table 7.1: Vulnerability Rubric

Score	0	1	4	7	10
	Attack has no effect on the trust in the system	Temporarily lowers trust in the system	Lowers trust in the system	Damages trust in the system	Destroys trust in the system
Detection	Is actively monitored for and prevented	Is monitored for and would be noticed on a daily report	Could take up to a week to recognize	Could take a month or more before it is recognized that the attack was implemented.	It is possible and likely that the attack would not be noticed until the attackers revealed their actions publicly (releasing a dataset, making public claims)
Damage	The attacker did nothing to the system that requires action	Requires minor code fixes to prevent future attacks	Requires alterations to the system and/or removal of the attackers code	The system requires significant alteration/repair	The system requires significant code updates and/or changes in the underlying protocol
	The attacker did nothing to individuals that requires action	The users are affected but no action is required	Some users will have to perform a one-time action to mitigate the effect	The attacker gained access to user information that cannot be altered easily	The repercussions from this attack will affect people on a continual basis and are difficult to quantify.

Summary of the Vulnerability Rubric Metrics

1. **Exploitability Metrics**. These are made up of characteristics of the vulnerable component and the attack against it
   (a) **Access**. Level of access (privilege) to the system required to perform the exploit
   (b) **Knowledge**. Level of knowledge required to implement the exploit
   (c) **Complexity**. What the technical requirements of the exploit are, the computing power, the workers required.
      i. **Technology**. The computing power required to perform or create the exploit
      ii. **Build**. The amount of workers it would take to build this attack and the time frame they could build it within
   (d) **Effort**. How much work it would take to operate the exploit
      i. **Planning**. The amount of components that have to work together to perform the exploit (that can't just be automated to perfectly work)
      ii. **Human**. The number of people required to work together to perform the exploit and whether they are aware of their involvement
2. **Impact Metrics**. These focus on the outcome that an exploit of the vulnerability could achieve
   (a) **Scope**. The amount of the user base that this exploit could affect
   (b) **Impact**. What the exploit allows an attacker to do, this can be get data logs, input their own data etc.
      i. **Data**. The kind of data that an attacker could get access to, how immediately useful it is to them. Or how far into the system they are able to place false data.
      ii. **Trust**. How this exploit being implemented will impact the user base's willingness to use the system, or a similar system in the future.
   (c) **Detection**. How easy it would be for someone to realize that this exploit has been used on the system
   (d) **Damage**. How easy it would be to fix what the attacker did
      i. **System**. How much the system caretaker needs to do to their system to have it safely operating again as it was before the attack.

ii. **User**. How the exploit would effect the individuals who were impacted by this in the long run

7.3.2 Attack Tree for Assessing Contact Tracing Application Vulnerability

To layout the possible vulnerabilities of the system, an attack tree was created. Attack trees are a formal and methodical way of describing the security of systems, based on various attacks. Attacks against the system are represented in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes [140].

In the attack tree built to layout possible avenues that could be used to take advantage of a contact tracing system, there are two possible goals. Either the attack breaks the privacy promises of the system, or the attack introduces false information into the system. These goals represent the different motives someone attacking a government system designed to protect the populace might have. Either they want information or they want to be mischievous and create panic or distrust.

The full attack tree is figure 7.1. The arrows represent the direction that the attack paths flow. The tree lays out 17 different avenues of attack. However, technically 18 will be posed against the chosen five systems. This is because for attack 4 it was pertinent to differentiate the attack between if someone tried to do it on their own and if some larger body wanted to try and perform it. Thus, there is an attack 4 and attack 4.5. All of the attacks are laid out below.

1. Capture your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

    - 4.5 Everything is the same as attack 4 except the attack is a company/organization that already has security cameras in their facility and is just adding the BLE receivers to the building

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy

7. Access a WiFi network → Capture messages between app and server → See IDs being sent from user to server indicating Covid+ status → Determine that person has the virus → Break privacy

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results

→ Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

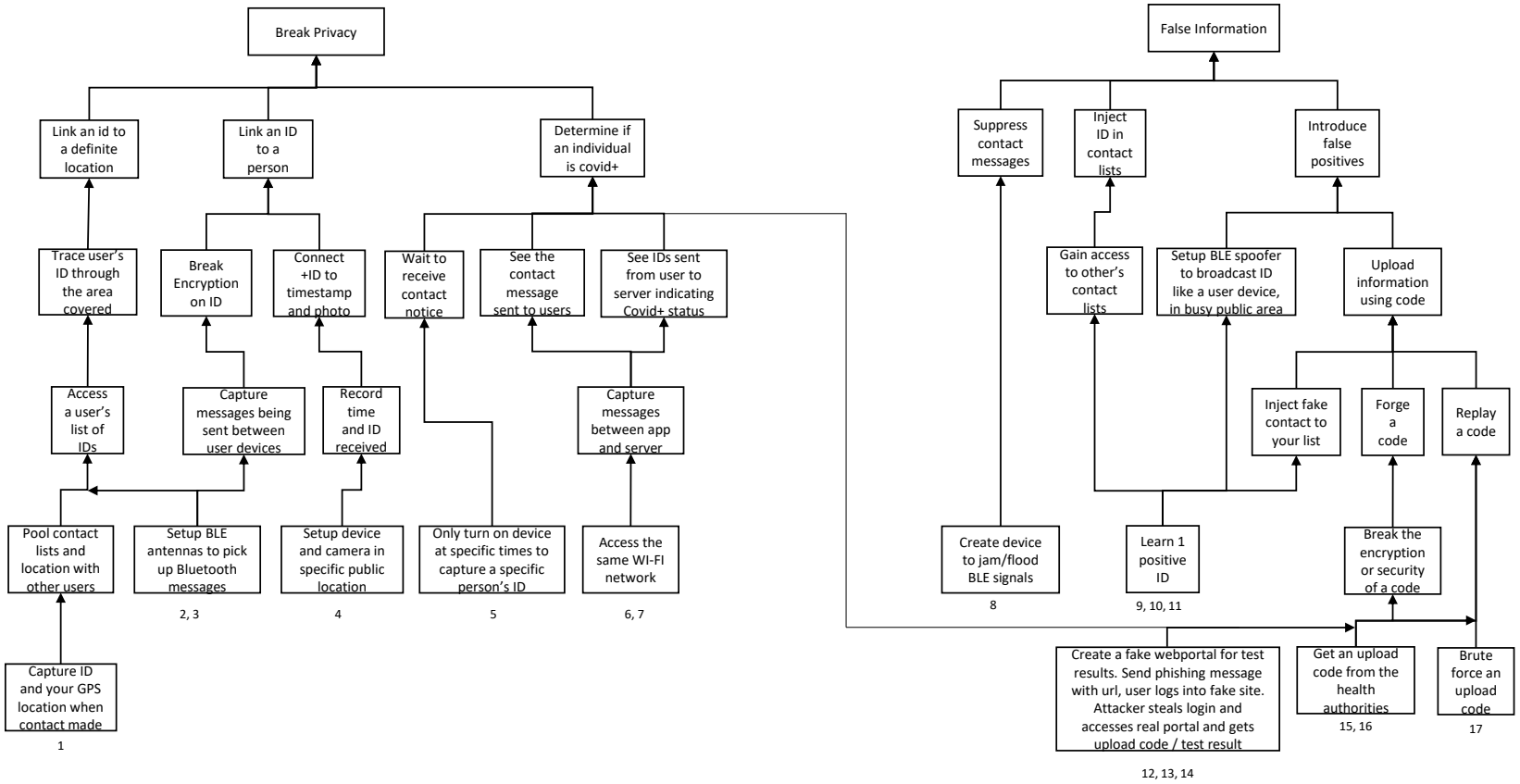


Figure 7.1: Attack tree created to represent possible avenues of malicious exploitation of digital contact tracing

Chapter 8

Assessing Contact Tracing Applications

8.1 Privacy Review of Contact Tracing Applications

For quick reference, a list of what constitutes a yes, half, or not met for each principle is listed below. Table 8.1 contains the review of fifty-five apps representing fifty different countries.

- Independent Expert Review:
  - Yes = The app was reviewed by independent experts who provided a documented review of the app prior to the release of it.
  - No = There is no clear independent review process performed or able to be performed
  - Half = There is the potential that it can be reviewed by independent experts, eg. the source code being available.
- Simple Design
  - Yes = Based on a publicly available white paper's protocol or design
  - No = No documentation of design released for review
  - Half = Some information about the design is available
- Minimal Functionality
  - Yes = Only does what is required for tracking based on protocol
  - No = There are a lot of extra functions

  - Half = Has 1 or 2 extra things that do not require extra data and relate to contact tracing

- Data Minimization
  - Yes = No personal information collected
  - No = Detailed health information, national IDs, detailed GPS data all collected
  - Half = Collects one or two pieces of information for clear contact tracing not identifying purposes

- Trusted Data Governance
  - Yes = Data is owned by the government department, does not leave the country, will not be accessed by any other body for any other purpose. Trustworthy actor subject to public oversight.
  - No = No clear ownership, or has been or could be used by other groups
  - Half = Ownership stated but missing the assurances

- Cybersecurity
  - Yes = Secure servers in the country, encrypted data in transit and storage, audits and monitoring to prevent and contain breaches
  - No = Security does not meet standard practice
  - Half = Implied security, or basic statements towards it

- Minimum Data Retention
  - Yes = 14 days as stated for the infectious period by WHO
  - No = Unclear or over years
  - Half = Longer than WHO stated infectious period but a stated length less than a year

- Protection of Derived Data and Meta-Data
  - Yes = Stated that the data is not created, or is deleted, or is otherwise secured
  - No = Not mentioned, or information is given to a third-party

  - Half = Not specifically mentioned but some information about it, eg. deleting IP addresses from server logs

- Proper Disclosure and Consent

  - Yes = App use is voluntary, all data releases are voluntary, app can be deleted or turned off. Privacy policy is easy to find detailed and clear.
  - No = App is mandatory to use or data is automatically released
  - Half = there is a privacy policy, but not everything that the app uses is explicitly mentioned. Or privacy related information was found but there was no clear privacy policy document.

- Provision to Sunset

  - Yes = Stated to sunset and clear method for determining when and how
  - No = Not mentioned or implied to continue beyond health crisis
  - Half = Stated to sunset but unclear as to definite time or method to determine

8.1.1 Discussion of the Privacy Review

A lot of information about the different applications was disseminated to review their privacy. A list of the resources used to collect this information is available in Appendix A. While every detail of every app will not be discussed here some information on various apps as well as comments on the decisions that were made will be discussed.

Many of the apps had various additional functions. For example, in the UK their app "NHS COVID-19" uses the GAEN framework for Bluetooth ID exchange but also has a QR code venue logging system [115]. This allows users to scan the QR code of a venue into their app and if there was an outbreak at that venue receive an alert if they may be at risk. The NHS app also has a map showing the risk levels of areas in England, a symptom tracker, a booking system for tests, and a self-isolation tool [115]. These types of functions are fairly representative of the types of functions most other apps feature. However, the contact tracing app released in Moscow Russia has a unique method of proving location. For the "Social Monitoring Service" app users will randomly receive push notifications requesting users to take a selfie (a picture of themselves) in the app [165]. If the user does not take the picture within the time limit they are automatically fined. In some cases, the notifications have been received in the middle of the

Table 8.1: Summary of App Privacy

Country	App	Independent Review	Simple Design	Min Functionality	Data Minimization	Data Governance	Cybersecurity	Min Data Retention	Protection of Meta-data	Disclosure and Consent	Provision to Sunset
Australia	COVIDSafe	◑[1]	●[7]	●	◑[28]	●	◑[62]	◑[69]	◑[72]	●	◑[80]
Austria	Stopp Corona	◑[2]	◑[8]	●	◑[29]	○[51]	●	◑[69]	○[36]	●	○[36]
Azerbaijan	e-Tabib	○[36]	○[36]	○[36]	◑[30]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]
Bahrain	BeAware Bahrain	○[36]	○[36]	○[36]	○[31]	○[52]	○[36]	○[36]	○[36]	○[36]	○[36]
Bangladesh	Corona Tracer BD	○[36]	◑[9]	●	◑[30]	◑[53]	◑[63]	○[36]	○[36]	○[36]	○[36]
Canada	Covid Alert	●	●[10]	●	●	◑[54]	●	●	◑[73]	●	◑[80]
China	Health Code	○[36]	○[11]	●[15]	○[32]	●[55]	○[36]	○[36]	○[36]	○[75]	○
Colombia	CoronApp	○[36]	○[9,11]	◑[16]	○[33]	◑[51]	○[36]	○[36]	◑[72]	◑[76]	◑[80]
Czech Republic	eRouška (eFacemask)	●	◑[8]	●	◑[30]	◑[56]	○[36]	◑[69]	●	●	○[36]
Denmark	Smittestop	◑[3]	●[10]	●	○[34]	◑[56]	○[36]	●	◑[72]	●	○[36]
Ecuador	ASI (SO)	◑[2]	●[10]	◑[17]	◑[35]	◑[53]	○[36]	◑[69]	○[36]	○[36]	○[36]
Estonia	Hoia	◑[2]	●[10]	●	●	●	○[36]	●	○[36]	●	○[36]
Ethiopia	Debo	○[36]	○[9]	◑[18]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]
France	TousAntiCovid	◑[2]	●[12]	●	○[36]	○[51]	◑[64]	●	◑[72]	◑[77]	◑[80]
Fiji	careFIJI	○[36]	●[7]	●	●	○[36]	◑[64]	◑[69]	○[36]	●	○[36]
Finland	Koronavilkku	◑[4]	●[10]	◑[17]	●	●	◑[63]	○[70]	○[36]	◑[77]	○[36]
Germany	Corona-Warn-App	●	●[10]	◑[19]	●	●	◑[65]	●	●	●	○[36]
Ghana	GH Covid-19 Tracker App	○[36]	○[36]	○[20]	◑[30]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]
Gibraltar	Beat Covid Gibraltar	○[36]	●	●	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]
Guatemala	Alerta Guate	○[36]	○[36]	○[21]	○[37]	○[57]	○[36]	○[71]	○[36]	○[36]	○[36]
Hungary	VirusRadar	○[36]	○[9]	●	◑[30]	○[36]	○[36]	●	○[36]	◑[77]	○[36]
Iceland	Rakning C-19	◑[1]	●[13]	●	○[38]	●	●	●	○[36]	●	○[36]
India	Aarogya Setu	●	●[9,13]	○[22]	○[39]	○[36]	●	◑[69]	○[36]	○[78]	◑[80]
Ireland	COVID Tracker	◑[2]	●[10]	◑[17]	◑[40]	◑[56]	●	●	●	●	○[36]
Israel	HaMagen	●	○[9,13]	◑[23]	○[41]	◑[53]	◑[63]	●[71]	○[36]	●	○[36]
Italy	Immuni	◑[2]	●[10]	●	●	●	●	◑[70]	○[36]	◑[77]	○[36]
Japan	COVID-19 Contact Confirming Application	◑[2]	●[10]	●	●	◑[53]	●	●	○[36]	●	○[36]
Jordan	AMAN APP - Jordan	○[36]	○[13]	●	○[42]	◑[53]	◑[64]	●	○[36]	●	○[36]
Kasakhstan	eGovbizbirgemiz mobile app	○[36]	●[10]	◑[19]	●	◑[53]	○[36]	●	○[36]	●	○[36]
Kuwait	Shlonik	○[36]	○[13]	◑[17]	○[43]	○[51]	○[36]	○[36]	○[36]	○[36]	○[36]

Table 8.1: Summary of App Privacy

Country	App	Independent Review	Simple Design	Min Functionality	Data Minimization	Data Governance	Cybersecurity	Min Data Retention	Protection of Meta-data	Disclosure and Consent	Provision to Sunset
Latvia	Apturi Covid	◑[1]	●[10]	●	◑[30]	○[51]	○[36]	●	○[36]	●	○[36]
Malaysia	MyTrace	○[36]	○[9]	●	◑[30]	◑[53]	◑[66]	◑[69]	○[36]	◑[77]	○[36]
Netherlands	CoronaMelder	◑[2]	○[9]	●	●	◑[53]	◑[66]	●	●	●	○[36]
New Zealand	NZ COVID Tracer	◑[5]	○[14]	●	◑[44]	◑[58]	◑[66]	○[70]	◑[74]	●	○[36]
North Macedonia	Stop Korona!	○[36]	○[9]	●	◑[30]	○[36]	○[36]	●	○[36]	●	○[36]
Northern Ireland	StopCOVID NI	◑[2]	●[10]	●	◑[30]	◑[59]	●	●	◑[72]	●	○[36]
Norway	Smittestopp	○	○[13]	●	○[42]	○[36]	○[36]	○[36]	○[36]	◑[77]	◑[81]
Poland	ProteGO Safe	●	●[10]	◑[17]	◑[30]	○[36]	◑[64]	●	○[36]	◑[77]	○[36]
Portugal	STAYAWAY COVID	◑[4]	●[10]	●	●	◑[60]	○[36]	●	○[36]	◑[77]	◑[80]
Qatar	Ehteraz App	○[36]	○[11,13]	○[24]	○[34]	○[36]	○[67]	○[36]	○[36]	○[79]	○[36]
Russia (Moscow)	Social Monitoring Service	○[36]	○[13]	◑[25]	○[45]	◑[51]	○[36]	○[71]	○[36]	○[79]	○[36]
Russia	Contact Tracer	○[36]	○[9,13]	○[26]	●[46]	○[36]	○[36]	○[36]	○[36]	○[36]	○[36]
Saudia Arabia	Tabaud	○[36]	●[10]	◑[17]	○[47]	◑[53]	◑[67]	●	○[36]	●	○[36]
Scotland	Protect Scotland	●	●[10]	●	◑[30]	◑[59]	◑[62]	●	○[36]	●	○[36]
Singapore	TraceTogether	◑[6]	●[7]	●	○[48]	◑[53]	◑[62]	◑[69]	○[36]	●	◑[80]
Slovenia	#OstaniZdrav	○[36]	●[10]	●	●	●	◑[68]	●	○[36]	●	○[36]
South Africa	COVID Alert South Africa	○[36]	●[10]	●	◑[49]	◑[53]	○[36]	●	○[36]	●	○[36]
Spain	Radar COVID	◑[2]	●[10]	●	●	●	◑[66]	●	○[36]	●	○[36]
Switzerland	SwissCovid App	●	●[10]	●	●	◑[59]	○[36]	●	○[36]	●	◑[80]
United Kingdom	NHS COVID-19	●	●[10,14]	◑[27]	◑[50]	○[36]	○[36]	◑[69]	○[36]	●	○[36]
United States	CoEpi	◑[2]	○[9]	◑[17]	●	○[36]	○[36]	○[36]	○[36]	◑[77]	○[36]
United States	Covid-Safe/CommonCircle Exposures	◑[1]	●[10]	●	●	○[36]	◑[64]	○[36]	○[36]	◑[77]	○[36]
United States (Arizona)	Covid Watch	◑[2]	●[10]	●	●	○[36]	○[36]	◑[69]	○[36]	◑[77]	○[36]
United States (California)	California COVID Notify	○[36]	●[10]	●	●	○[36]	○[36]	●	○[36]	●	○[36]
United States (North Dakota and Wyoming)	Care19 Alert	○[36]	●[10]	◑[23]	○[42]	◑[61]	○[36]	●	○[36]	●	○[36]

[1] Released the source code for the app side but not server, no mention of review performed prior to release
[2] Released the source code for entire system, no mention of review performed prior to release
[3] Government agency performed assessment, specifically does not release source code to protect the app from hackers
[4] Released the source code for entire system, government agency performed assessment
[5] Government agency performed assessment
[6] Open-source code for the entire system, 4 independent experts consulted for use of the tokens no mention of assessment for app or system
[7] Based on Bluetrace protocol and Open trace open-source code base available online for review
[8] Future update will employ GAEN and DP-3T protocols
[9] Bluetooth based design but no white paper released on the specific protocol being employed
[10]Using the GAEN API
[11]QR code digital pass system but no white paper released on the specific protocol being employed
[12]Bluetooth based design following ROBERT protocol
[13]GPS location tracking but no white paper released on the specific protocol being employed
[14]QR code location based logging but no white paper released on the specific protocol being employed
[15]Information is being sent to the server continuously and from sources other than the user's app
[16]Information entered into the system is sent with GPS data for hotspot analysis
[17]Also has a personal symptom tracker built in
[18]Also has a manual contact tracker built in to log contacts you know
[19]Also provides access to test results
[20]App is also a symptom tracker, hotspot identifier, event management aid, and quarantine manager
[21]App is predominantly a emergency alert system that has had contact tracing added to it
[22]App has bluetooth token exchange, GPS logging, a symptom assessment, displayed risk status of user, location based covid statistics, e-pass integration, time and status of contacts recorded
[23]App uses both bluetooth and GPS for contact logging
[24]App requires access to files that does not make sense for the proposed design
[25]App requires selfie (picture) based location confirmation
[26]App has bluetooth token exchange, GPS logging, a symptom assessment, and location based covid statistics
[27]A symptom assessment, venue logging using QR codes, area based alerts, Covid test booking, quarantine aids
[28]A name (can be pseudonym), phone number, age range, postal code
[29]A phone number when user voluntarily informs the system of a positive status
[30]A phone number upon registration
[31]Location data that is actively uploaded
[32]GPS data, travel history, health information, medical test results, spending history, and phone numbers which in China these are in a database tied to government IDs
[33]GPS data, health information, phone number, age, sex, ethnicity, email, national ID, previous movement history, and requests information on participation in mass events
[34]National ID
[35]Area user lives to the municipal equivalent level
[36]No information was found
[37]Email address or social media account at registration, collects location data
[38]GPS data (14 days), requires phone number, if user is positive requests national ID number
[39]Phone number, name, gender, age, profession, travel history for last 30 days, willingness to volunteer in times of need, displays information about nearby users
[40]Phone number, county or town, age group, and sex
[41]GPS data (14 days), history of connected wireless networks
[42]GPS data (14 days)
[43]GPS data, phone number, civil ID
[44]Email address, phone number, National Health Index Number
[45]GPS location, app accesses on the device calls, camera, network information, sensors
[46]GPS location, displays information about nearby users
[47]National ID, date of birth, phone number

[48]National ID, phone number
[49]Date of birth
[50]Half postal code, venue time and data, isolation status
[51]Location of server unknown, no assurances of data access limitations or oversight
[52]Televised game show gives cash rewards to randomized users that are at home when their name is selected, real-time tracking of quarantined individuals provided to health workers
[53]Data controlled by government authority, the location of the server is unknown
[54]Server located within country, data ownership unknown
[55]User's location and identifying code number is given to a police server
[56]Government authority administrator of data, unknown location of server, third-party used for data processing
[57]App developer had access to user data, third-party vendors collect, store, and process data
[58]Government authority administrator of data, known location of server outside of country, limitations on data usage
[59]Government authority administrator of data, server within country, third-party used for data processing
[60]Server inside of country, unknown government authority acting as administrator of data
[61]Data ownership is known but not governmental server is owned by a third-party who also processes some of the data
[62]Data is encrypted when stored, no information about in transit etc.
[63]Data is encrypted in transit, no information about storage etc.
[64]Data is encrypted on device, no information about in transit etc.
[65]Data is encrypted in transit, QR code required to retrieve test results, random noise will be added to notifications on the server side
[66]Assurances that everything will be to the government's security standards, no specifics
[67]No specifics, previously a security flaw was found that exposed sensitive user data
[68]Metadata sent to the server is encrypted
[69]Stated data retention limit longer than WHO infectious period of 14 days
[70]Stated data retention limit longer than WHO infectious period of 14 days on the device and held on the server until no longer needed
[71]Stated data retention limit is years in length
[72]Stated to meet relevant Privacy laws in country
[73]Stated to be designed to minimize data generated/collected
[74]Stated to de-identify any of the information collected for statistical purposes
[75]There are negative impacts on a users daily life if they choose to not use the app or disclose data
[76]The app may be required to access necessities
[77]A privacy policy was not found
[78]Government previously made it mandatory for any one employed, there is no required consent to upload information
[79]Government has made use of the app mandatory
[80]Unclear as to definite time, stated as when a health authority declares the pandemic over
[81]App was already removed from the app store due to security concerns

night when the user was asleep, or in one case the user was in an ambulance on the way to the hospital [165].

For data minimization, there were some cases, such as the Australian "COVIDSafe" app, that toes the line of information collection. The Australian app collects a name (which can be a pseudonym), age range, mobile phone number, and postal code [64]. All of this information has a reason to be collected however it was discussed in the de-identification review of 3 and 4 that it is possible to identify someone from their age and postal code. Also, a phone number can easily be an identifier because people can look up the owner of the number.

Of course, there are also apps such as Bahrain's "BeAware Bahrain". The app is actively uploading the GPS information of its users to a server. It notifies users if they are nearing an area that currently has active cases as well as alerting authorities if someone who should be under home quarantine leaves their home [117].

In Colombia, they released an app called "CoronApp". Within the app, users are asked whether they have attended any large public gatherings. Due to the recent protests in Colombia, there is the suspicion that the question is being used to find protesters [69]. The system also asks for names, ages, identification numbers, symptoms, prior medical history, and previous movements [41].

For the principle of data governance, there are a few notes that should be made. First is that it is possible that European countries are following the General Data Protection Regulation (GDPR) guidelines and that those entail an amount of data governance that is overlapping with what the principle requires. However, if they are marked as not meeting data governance then from the findings of the review they did not state that they were following GDPR and what that entails. As part of the disclosure requirement, it should not be expected of users to go through the legalize of this document and determine what exactly the authority is doing. As well in Iceland, the data remains within the EU though it does not specify if it remains in Iceland [123]. However, because of the nature of the EU Iceland has been treated as though the information remains within the country. Similarly the data server for the Scottish app "Protect Scotland" has servers in London England [141]. Scotland and England are both a part of the United Kingdom thus Scotland was treated as though the information remained within the country.

An app with very interesting data governance is Bahrain. As an incentive to get more citizens to download the app users were entered into a game show [9]. The game was that the host would randomly call a user of the app and if they were at home obeying the lockdown

rules they would receive 1,000 Bahraini dinars (approximately CAD 3475.88). The lockdown game show called “Are you at home” is put together by Bahrain TV. Originally users were automatically entered into the game show, this was altered so that users have the option to opt-out. [9]

The cybersecurity principle only had 8 apps meet the requirement. It is interesting to compare how different countries approach cybersecurity. In India, the government has set up a bug bounty program for their app [119]. Though some of the criteria of what they will accept as a vulnerability are strict and specific as some bounty seekers attest [1]. There are also many countries such as Canada, Singapore, Australia, India, Iceland, the Netherlands, and others that released some aspect of their source code. Then there are countries such as Denmark, whose security stance is that they will not be releasing any source code as doing so would make the system less secure against attackers [118]. The statement implying that security through obscurity is preferred. This is directly opposite to Kerckhoffs’s principle.

The ruling for minimum data retention is based on the 14 day infection period stated by the WHO, Canada retains the ID logs for 15 days rather than 14 days [21]. It was determined that this single day difference was close enough to be accepted.

In some cases the data that was held the longest was used as the determining factor for instance in the Czech Republic the information about contacts that is sent to the server is only held for twelve hours, however, the contact list on the device is stored for 30 days [124]. Israel stores data on the device for 14 days but data on the server including information irrelevant to contact tracing will be held for 7 years [121]. Whereas in Iceland the only data held for longer than 14 days was the phone number of the user. This is stated to be because the Icelandic system requires the phone number to notify users if they have had a contact [123]. Thus this situation still passed the requirement. As did Ireland’s system which holds everything for 14 days except the symptom log that it holds for 28 days [45]. This was permitted because the symptom log does not leave the device and while the infectious period is listed as 14 days the symptoms from the virus could last longer than that [101].

In Finland the legislation currently allows data to be held until March 2021 when the system will be reassessed [120]. Finland and other countries that have similar data hold lengths did not pass the requirement because the length of time is beyond a year and could be extended. Other data clauses specify that the data will be held until no longer needed. Typically it is unclear what criteria will be used to determine that the data is no longer needed.

In New Zealand location data is held for 60 days on the device, and personal information will be held the length of the pandemic on the server. Also, some of the information from the app can become part of a user's permanent health record [122].

From the privacy review, it is clear that the two principles that the fewest apps meet are Protection of Derived Data and Meta-data and Provision to Sunset. Only four of fifty-five apps meet the meta-data principle requirement and all forty-four of the apps that do not meet the criteria are because of a lack of information.

Some countries like Australia [64] or Denmark [118] have stated that their app meets the privacy requirement of applicable law. However, as noted before with the GDPR in Europe as part of the disclosure requirement it should not be expected of users to go through the legalize of a document and determine what exactly the authority is doing. The authority should state what protections they are providing.

As mentioned in section 7.2.2 the definition of voluntary to use is not just that using the app is not mandated by the authority but that an individual's life is not negatively impacted if they choose not to use the app. For example in China if you do not disclose the information the app asks for it will not provide a code to the user. A green code in the app is required to use public transit and enter many public spaces [108].

In the city of Medellin in Colombia, it is being proposed that access to essential services like grocery stores should require an individual to display the QR code from the app [41]. In Qatar, it was made mandatory for citizens and residents to have the app "Ehteraz" active on their phone when leaving the house. Not having the app could lead to a maximum fine of $55,000 or three years in prison [74]. The Moscow app is mandatory for anyone who shows symptoms of COVID-19. Anyone not complying can be penalized with fines or forced hospitalization [165].

Provision to sunset has nine apps that half meet the requirement, one that does not meet the requirement and the rest of the forty-five do not meet the requirement because of a lack of information. It is a concern that so many governments are releasing these apps without giving the public information on if they are planning on continuing to use these apps after the current pandemic.

The only app to not pass the sunset requirement based on information available was the " Health Code" app in use in China. Though the authorities have not stated outright that they intend to continue using the app beyond the current health crisis there is evidence to suggest that the system could continue to be used beyond the life of the pandemic [15].

There were some apps that were discovered over the course of the research however not enough information on them could be collected for their inclusion in the review to be meaningful. These included Croatia's "Stop COVID-19", Nepal's "COVIRA App", South Korea's system, and Turkey's "Life Fits Inside the House". The Norway app "Smittestop" was removed from the app store after security issues were brought to light [90].

8.1.2 Selected Applications for Analysis

The five apps chosen to analyze closely were Canada's Covid Alert App, Singapore's Trace-together app, Iceland's Rakning C-19 app, India's Aarogya Setu app, and France's TousAnti-Covid app. These were chosen as representatives of the different systems of contact tracing. Canada's Covid Alert app is a decentralized Bluetooth System implemented using the GAEN Framework. Singapore's TracerTogether app is a centralized Bluetooth system following the BlueTrace protocol. Iceland's Rakning C-19 app is a centralized GPS system. India's Aarogya Setu is a Bluetooth centralized system that has added functionalities involving location data. France's TousAntiCovid is a centralized Bluetooth system that follows the ROBERT protocol.

Three of the apps use the Bluetooth centralized system because this was the model that had the most variance. The large majority of the decentralized Bluetooth systems used GAEN and were similar apps with slight differences between them. The Canadian app does not have any added functionality. Iceland's Rakning C-19 app was chosen because it is a GPS system that had a fair amount of information released about it as they have the source code available online. Singapore, India, and France represent some of the differences in system and functionality that appear between the apps.

8.1.3 Privacy Ranking of Selected Applications

Canada's Covid Alert has a lot of information available about their app, the privacy statements are detailed and clear. All of the principles are addressed, though they were not all completely met. In terms of privacy, Covid Alert was grouped as Green.

Singapore's TraceTogether does have a lot of information available but does not address all of the principles. The app also requests personal information that is not required for their contact tracing protocol to operate. TraceTogether was grouped as Yellow.

Iceland's Rakning C-19 was difficult to group. Though there is a lot of information available about the app and they do meet quite a few of the principles there are still some that they do not meet. As well, GPS data is, as noted in the deidentification review, a lot of personal information that is difficult to remove someone's identity from. If in the future it could be shown that GPS systems are significantly better than Bluetooth at alleviating the health crisis then the risk to reward could be reassessed but for now, there is no evidence of that, and GPS data is high risk. Rakning C-19 was given a Red/Yellow ranking. Meaning that either the red or yellow group would be accepted as the thresholds are determined.

India's Aarogya Setu did not meet the most principles of the apps selected. This app is interesting in that it is a Bluetooth system but it also takes some location data from the user and displays that to other users showing them how many people at risk they were near. As well the system does not ask for consent for uploading information. Once the app is downloaded if a user tests positive their account on the server is flagged and the next time their device connects it uploads the information required. Due to these factors, the Aarogya Setu app was given a Red grouping.

France's TousAntiCovid has a lot of information available about the app. However, the data governance principle is not met and it is unclear what personal information the app requests from the user. It also half meets more principles than it completely meets. As it stands the TousAntiCovid app was grouped as Yellow.

A summary of the groupings of the five apps:

- Canada Covid Alert - Green
- Singapore TraceTogether - Yellow
- Iceland Rakning C-19 - Red/yellow
- India Aarogya Setu - Red
- France TousAntiCovid - Yellow

A quantitative analysis of the data from the privacy review table 8.1. Each principle will be given a score out of 1. If the app meets the principle it will be given a 1/1. Half meeting the

Table 8.2: Analysis of Privacy Review

Principles	Covid Alert	TraceTogether	Rakning C-19	Aarogya Setu	TousAntiCovid
Met	7	3	6	3	3
Half-met	3	5	1	2	5
Not Met	0	2	3	5	2
Total:	8.5	5.5	6.5	4	5.5

principle gives a 0.5/1. Not meeting the principle gives a 0/1. Thus the privacy of an app can be scored out of 10. The scoring of each app can be seen in table 8.2

Using this method of ranking a red app is one that scores 0-4.5, a yellow app is 5-7.5, and a green app is 8-10. This means that an app has to meet five principles, half of all the principles or some combination to be outside of the red group. Scoring less than 50% on the principles is a concern. An app scoring less than 50% should be a concern because it can be seen as the developer or the authority not prioritizing privacy in the design of their system. This is a concern because privacy by design, or creating the design with privacy in mind from the beginning is the best way to implement privacy throughout the design [25].

This ranking method does not prioritize any single principle, instead it takes them as equally important. Something that is reflected by the author's presentation of them. There was never any indication that the order in which they were presented was meaningful. If one were to attempt to form an internal or weighted ranking of the principles an argument could be made for any one of them to be key or very important. Thus, leaving them as equal maintains the interpretation that they are all important to the overall privacy of the system.

8.2 Analysis of the Vulnerability of Contact Tracing Applications

8.2.1 Canada Covid Alert

Attack 1

1. Capture your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 2** There is nothing in the protocol to prevent someone from building a tool to collect GPS location data when the phone makes a BLE connection. The attack requires either circumventing the app to log the contact information outside of it or rooting the attackers device to access the logged contact information. The bottle neck of attack 1 is getting the temporary IDs of the victim to trace through the system. In the Covid Alert documentation all it says is that the temporary IDs are "securely stored" without a clear definition of what that means [128]. It is assumed that to see the temporary IDs the attacker would need to root the victims device. Thus this attack should be marked as a 1. However, because the IDs of positive users are made available through the server an attacker could collect all of this information then recreate the list of IDs of positive users find matches and follow them. This possibility is why the score is a 2.

- **Knowledge: 3** The attack would require being able to make the GPS logging code, detect that the device has logged the Bluetooth information. Root the attacker's own device and potentially the victims, or pull the list from the server and recreate the IDs of positive users.

- **Complexity: 6**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The analysis can be done on a consumer computer.

  - **Build: 5** The attack requires one or two tools, the GPS logger, the access to the contact list, and the method of collecting the IDs to track. Then the program to match the ID and every GPS coordinate and map them together. One person could do this in more than a month, or a few could do this in about a month. All of these components have been created before for other attacks (the GPS data attacks of section 3.2.3) and would need to be put together for this one.

- **Effort: 1.5**

  - **Planning: 1** This is high effort because the code has to be placed on many phones to collect all of the required information. Then the data has to be compiled.

  - **Human: 2** To effectively cover an area quite a few people would need to be involved. Once the software is on their phone they would not have to necessarily do anything out of the ordinary however just go through their usual routine.

- **Scope: 4** This would affect the group IDs could be identified for. People close to any of those involved whose devices could be accessed to get the IDs or people who test positive in the community. This is smaller than everyone an attacker knows and is not every positive individual as it is relegated to the area covered.

- **Impact: 4**

  - **Data: 4** The contact log information itself is not easy to use, but the GPS data allows the creation of a map of where someone has been and could then determine where they live or something about them from that as seen in the attacks discussed in 3.2.3.

  - **Trust: 4** This would damage trust in the system. There is the potential that a politically motivated group could use this to single out certain people. Especially if they can target those that have the virus.

- **Detection: 8** It is possible that this would not be noticed, the code that collects the info does not need to interfere with the app, just monitor it, though the amount of people involved would require quite the vow of secrecy. It could involve access to someone's phone to retrieve the IDs which would increase the risk of detection.

- **Damage: 2.5**

  - **System: 1** The code added to the groups devices would have to be removed but otherwise the system itself has not truly been touched

  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of GPS location data discussed in section 2.2.

Average: 3.875

Attack 2

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 2** The limitation of attack 2 is getting the temporary IDs of the victim to trace through the system. In the Covid Alert documentation all it says is that the temporary IDs are "securely stored" without a clear definition of what that means [128]. It is assumed that to see the temporary IDs the attacker would need to root the victims device. Thus this attack should be marked as a 1. However, because the IDs of positive users are made available through the server an attacker could collect all of this information then recreate the list of IDs of positive users find matches and follow them. This possibility is why the score is a 2. The exploit also requires physical access to locations to place the BLE devices.

- **Knowledge: 3** An exploit would require creating the BLE devices and mimicking the Bluetooth transmissions of the GAENs protocol, or the attacker could just have the app on the device then send the message from that through the BLE antennas and boosters. The victims device would need to be rooted to gain the IDs, or the list published by the server could be used to recreate the IDs of positive patients. The attacker would need to also access their contact logs.

- **Complexity: 6.5**

  - **Technology: 7** The generation of someone's temporary ID is possible from a phone as that is what happens when the app downloads the positive keys and then checks against the contact list. The BLE devices would likely require a standard computer to setup.

  - **Build: 6** The creation of the the Bluetooth devices is something that a single person could do with enough time. Creating enough of them to cover a larger area would be time consuming, as would the actual setup of the devices in the locations.

- **Effort: 6**

  - **Planning: 5** Medium to low effort, the BLE components need to be set up and placed in the area. Retrieving the IDs from a victim's phone would require care. Though taking them off the server could remove this factor.

  - **Human: 7**. One or two people could do this, setting up the BLE devices might require two people.

- **Scope: 4** This would effect some people that the attacker knows personally at most. It would be difficult to get someone's temporary ID that you did not know off of their device. Though the attack could target people that later test positive, the amount that had also gone through the targeted area would be limited.
- **Impact: 4**
  - **Data: 4** The attack provides information on everywhere an individual has been over an area covered by the BLE devices. Though this might not be able to be used to determine where they live there remains information that could be used to identify them. This was discussed in section 3.2.3.
  - **Trust: 4** Due to the promises of not being able to track people with the app this would make people wary about downloading the app.
- **Detection: 8** It is possible that this attack would not be noticed, the only time it could be would be when an attacker tried to take the ID if the attacker was trying to get it from the victim's phone.
- **Damage: 2**
  - **System: 0** Nothing directly damaging to the system has been done.
  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of GPS location data discussed in section 2.2.

Average: 4.438

Attack 3

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

This app is created using the GAENs framework. This means that the temporary ID that is broadcast is randomly created through cryptographic processes. Breaking the encryption as it were is beyond the scope of this thesis and would not yield information that could be used to determine a users identity. Thus this vulnerability path is infeasible.

Attack 4

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

To implement attack 4 the time of the contact needs to be known to the attacker to match that with the camera. The contact list stores the time stamp thus an attacker needs to access the contact list on their device. An added complexity is identifying who the person was from a photo. If an attacker were to setup a website with someone's photo saying that they have COVID-19 this could ID the person quickly, however it would immediately be a target for the authorities. At the very least the website would be taken down, and potentially information from that could be used to find the attacker. Thus doing so would make the attack significantly more detectable.

- **Access: 4** An exploit would require access to attackers own list of contacts and the list of positive IDs from the server.
- **Knowledge: 7** Other than accessing the contact log on the device performing the handshake this is not an attack that requires significant knowledge. A graduate level student could likely figure that out for their own device.
- **Complexity: 7**
  - **Tech: 7** In terms of computing power this can be done on a phone because the app performs the handshake on the phone. Though the matching of the contact list to the camera would require a standard computer.
  - **Personnel: 7** This should be possible for a single person to create, the difficult part being mimicking the blue tooth handshake or accessing the contact list of a device the attacker has control over. As mentioned previously this is not considered an impossible limitation.
- **Effort: 7**
  - **Planning: 7** low, the camera and the device logging the contact need to be setup. However the rest can be done once the data is collected and all that entails is matching the contact to the timestamp on the camera and then pulling information from the server to find a match.

  - **Human: 7** one or two people of this knowledge level would be able to create this and then go place it somewhere

- **Scope: 5** Affects a small group, as many people as an attacker could get to walk through a specific area.

- **Impact: 4**

  - **Data: 4** The attacker has gained information about who is actually sick, which is private health information, but they need to identify the people in the photo.

  - **Trust: 4** This would lower trust as the attack provides an attacker with the photo of someone who is sick.

- **Detection: 8** It is possible that this attack would not be noticed, the only time it could be would be if someone noticed the camera or if an attacker did something public to determine someone's identity.

- **Damage: 3.5**

  - **System: 0** Nothing directly damaging to the system has been done.

  - **User: 7** Someone has now had their privacy about health data stolen from them, which is private data that has many protections for a reason see section 2.2 for examples of why this data is protected.

Average: 5.688

Attack 4.5

The difference between attack 4 and 4.5 is who the attacker is. In 4 the attacker could be anyone, 4.5 looks at the specific case where an institution or organization of some kind is the one collecting the information. A system of BLE devices could be placed within a building or many buildings that the organization owns and already has a system of security cameras within. A workplace for example, a shopping mall, a campus. The group could already have the photo ID of the people that are regularly in the building on file if it is a workplace. They might just hold the data to be used for something at a later date or claim they wish to use it to ensure employee safety. The intention with this attack is to look at the long term implication of an organization hanging onto this data.

- **Access: 4**. An exploit would require access to attackers own list of contacts and the list of positive IDs from the server.
- **Knowledge: 7**. Other than accessing the contact log on the device performing the handshake this is not an attack that requires significant knowledge. A graduate level student could likely figure that out for their own device.
- **Complexity: 7**
  - **Tech: 7**. In terms of computing power this can be done on a phone because the app performs the handshake on the phone. Though the matching of the contact list to the camera would require a standard computer.
  - **Build: 7**. This should be possible for a single person to create, the difficult part being mimicking the blue tooth handshake or accessing the contact list of a device the attacker has control over. As mentioned previously this is not considered an impossible limitation.
- **Effort: 8.5**
  - **Planning: 7**. Though all of the BLE devices need to be created, the camera system is already available. The two systems require some work to put them together.
  - **Human: 10**. One person could set this up within the building or required area.
- **Scope: 6**. This wouldn't be a whole demographic, but would affect a large amount of people that are in that building or on company campus.
- **Impact: 7**
  - **Data: 7**. An organization with their own personnel files likely containing a photo would make identification easier.
  - **Trust: 7**. If the attacker is taking this data in this manner people may not find out about it for a long time however, when they do it will damage the trust in a future system like this.
- **Detection: 10**. The system has little to no way of detecting this. Even the victims would likely not notice anything as the security cameras were already there.
- **Damage: 5**

  - **System: 0**. The attack does not touch the system.
  - **User: 10**. They would have this information for a long time, they could use it in unexpected and interesting ways that could effect people for a long time without them even knowing it.

Average: 6.813

Attack 5

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

- **Access: 10**. The access of a regular user is all that is required for this attack.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. The only computer required is the phone itself.
  - **Build: 10**. A single person could create this in less than a month as there are no extra components required.
- **Effort: 10**
  - **Planning: 10**. Very low effort, there are few components or steps.
  - **Human: 10**. One person could perform this alone.
- **Scope: 1**. Could effect one person that the attacker knows. Then perhaps be used again. Long term the attacker could use this to effect a few more people.
- **Impact: 4.5**
  - **Data: 7**. The attacker gains knowledge of whether one person has the virus. This is protected health data.
  - **Trust: 2**. Though the scope is small, this is the kind of attack that could lower the trust of some vulnerable communities or people that are unlikely to trust the system to begin with.

- **Detection: 10**. It would be almost impossible for the system to catch that this has occurred.
- **Damage: 3**
  - **System: 0**. The attack does not touch the system.
  - **User: 6**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 7.313

Attack 6

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy

Due to the Covid Alert app using the decentralized Bluetooth system this attack is not possible. In the decentralized system the server does not send contact messages directly to users. This means there are no messages for an attacker to eavesdrop on in this manner.

Attack 7

The documentation for the Covid Alert app says that "Data at rest and in transit are encrypted using strong encryption methods" [129]. This statement has been interpreted to mean that communications between the server and the app are secured using HTTPS. Breaking HTTPS or Secure TLS is beyond the scope of this thesis. Thus it should not be possible for an attacker to eavesdrop on the communications as required by both of these avenues of vulnerability. However, analysis of the app has shown that it communicates with two different servers "retrieval.covid-notification.alpha.canada.ca" and "submission.covid-notification.alpha.canada.ca" [72]. The retrieval server is where the app receives the file of keys to compare to its contact log from. The submission server is what the app connects to when verifying a one-time code and uploading the user's IDs to the server. This means that the submission server is only connected to if a user has tested positive for COVID-19 and is informing the app's system. Thus, an eavesdropping attacker would still be able to see the server the user is communicating too and know that they have had a positive diagnosis by the server name.

- **Access: 7**. The attacker would need to be using a tool that provides them with information of what is being sent over the WiFi network. Many are available. They would also need to be in an area the WiFi network can reach, be physically present
- **Knowledge: 7**. Slightly more than novice level, this does not require particularly specific knowledge, but the attacker would need to be familiar with the tool they are using. There is a lot of software available that allows attackers to eavesdrop on a WiFi network. They would also need to know what part of the communication to grab the server name from, and have probed the servers to retrieve the names.
- **Complexity: 8.5**
  - **Technology: 7**. This attack only requires a computer.
  - **Build: 10**. A single person could create this attack in less than a month.
- **Effort: 6**
  - **Planning: 5**. The attacker needs to be in the right place at the right time to see the information being sent to that server. If they were able to find a clinic or somewhere that patients might receive the results of tests then they might increase the chances of catching this exchange. Such a thing would require a bit of effort to work out.
  - **Human: 7**. One or two people working together could perform this exploit.
- **Scope: 4**. The attacker is limited by having to be somewhere that an individual would be uploading this information from. Finding a location that a larger group of people would do this within would be difficult. Thus the scope is limited.
- **Impact: 4**
  - **Data: 4**. The attacker gains the information that a user on the WiFi has the virus. The attacker would need to determine who it was on the WiFi network that received the notification.
  - **Trust: 4**. As the authority assures the users that the system is private the ability for an attacker to learn that someone has the virus would lower trust in the system.
- **Detection: 9**. It is possible that this would not be noticed, the attack does not need to interfere with the app, just monitor the communications. The attacker physically being there would increase the risk of detection.

- **Damage: 3**
  - **System: 0**. The attack does not touch the system.
  - **User: 6**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.063

Attack 8

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

- **Access: 10**. The attack requires no system access, it only requires that something be placed in a physical location.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 9**
  - **Technology: 8**. This is possible with a device that would not have to be a computer, however could require a computer to build. Various devices can already prevent Bluetooth signals like microwave interference.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 7.5**
  - **Planning: 7**. To cover a large area an attacker would need multiple devices, as well as the placement of the devices to be discreet.
  - **Human: 8**. One or two people working together could create all of the device and place them in an area.
- **Scope: 6**. While not able to target an entire demographic a large area could be covered where anyone within would not receive any contact logs on their app.
- **Impact: 1.5**

  - **Data: 1**. The attack does introduce false data on the device, in that the device thinks that there are no devices close enough for a contact despite this not being the actual case. However, the data goes no farther than that into the system.
  - **Trust: 2**. This would cause people to ask why even use the app if the signals can be blocked. Temporarily lowering their trust in the system.

- **Detection: 4**. This attack would interfere with other Bluetooth devices in the same area, and potentially other signals such as WiFi. That makes this attack somewhat noticeable to anyone in the area.

- **Damage: 2**

  - **System: 0**. The attack does not touch the system.
  - **User: 4**. People in that area could have come into contact with the virus and not received the exposure notifications. A single test once the attack was discovered would let them know they are in the clear.

Average: 6.25

Attack 9

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

- **Access: 1**. For this attack the user would need to have root access on the victim's device, to be able to inject the ID into their contact list.

- **Knowledge: 2**. To gain root access on the victim's device a close to expert level of knowledge would be required.

- **Complexity: 5.5**

  - **Technology: 7**. Such an exploit would require a computer.
  - **Build: 4**. Finding a way to access the contact list of someone's device and injecting a contact that appears real would require a significant investment of time.

- **Effort: 7**

  - **Planning: 4**. There are multiple components that would have to come together for this exploit to work. Most importantly getting access to the victim's device.
  - **Human: 10**. After everything has been built only one person would be required to perform the attack.

- **Scope: 1**. It would be very difficult to attack one person's device in this way. A large target of people would require remote root access to their devices to be practically implementable. Otherwise the attacker has to have physical access to their target's phone.

- **Impact: 1**

  - **Data: 1**. The attack introduces false data onto the device that would then notify the user that they have had an exposure. The information goes no farther into the system.
  - **Trust: 1**. The scope of the attack is small, and while it is inconvenient to the victim to have to quarantine or get tested. The attack would only temporarily lower trust in the system.

- **Detection: 8** It is possible that this would not be noticed, it does involve access to someone's phone to inject the ID which is the most likely point of detection.

- **Damage: 2**

  - **System: 0**. The attack does touch the system but the contact logs are removed after 15 days.
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 3.438

Attack 10

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

- **Access: 4**. The attacker needs to pull the positive ID list down from the server which is outside of the typical user privilege. As is creating the Bluetooth message being transmitted.

- **Knowledge: 4**. An attacker needs to be able to recreate the processes used to generate the IDs to turn the information from the server into what would actually have been broadcast by a device. They also need to be able to recreate the entire bluetooth message, and have their devices operate as though they are the regular app. This indicates a knowledge level of someone with a career in this area.

- **Complexity: 7**

  - **Technology: 7**. Such an exploit would require a computer.
  - **Build: 7**. One person could create this attack in a month.

- **Effort: 6.5**

  - **Planning: 6**. Medium level of effort, there are several components, and the setup of the BLE devices in the target area.
  - **Human: 7**. One or two people could do this, setting up the BLE devices might require two people.

- **Scope: 6**. While not able to target an entire demographic a large area could be covered where anyone within would have a false positive contact logged.

- **Impact: 2**

  - **Data: 1**. The attacker introduces false data on the device, that will be deleted after 15 days.
  - **Trust: 3**. This attack would lower trust in the system. The victims would all be instructed to quarantine and test, something that inconveniences their lives and would make them less likely to trust the next notification they might get.

- **Detection: 8**. The system itself would not notice the attack, and users are not informed of the time or location of the contact they are being warned about. So the only time it could be noticed is if over time enough people that had all been to the same place, received the notification, and the negative test result and someone within the system recognized

this. Which could take longer than a week but is possible without the attacker informing anyone.

- **Damage: 2**
  - **System: 0**. The attack does touch the system but the contact logs are removed after 15 days.
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 4.938

Attack 11

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

The Covid Alert system requires more than just an exposure notification for a user to be able to upload their information to the server. A one-time code only received through a positive test must be entered into the app to upload the information. Thus, this potential avenue of vulnerability is not viable for this system.

Attack 12

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

- **Access: 9**. The attacker does not need to have access to the system or the app but does need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive
- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.
- **Complexity: 7**

  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.
  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.

- **Effort: 7**
  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.
  - **Human: 7**. One or two people working together would be able to implement this attack.

- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.

- **Impact: 7**
  - **Data: 10**. The attacker would have the health information of the victims
  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.

- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. Most email or other systems have filters that will catch a lot of phishing attempts.

- **Damage: 3.5**
  - **System: 0**. The attack does touch the system.
  - **User: 7**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.438

Attack 13

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information in the documentation for Covid Alert on how the one-time verification codes are created. It does however state that they are random. There is information in the documentation that the code is verified when entered [22]. Such verification would be a simple task of comparing the code entered to a valid code list. This is a very likely method of verification because the system only requires a small number of codes to be active at a time. Thus, forging a code would be blocked by the verification of the code, and this path is not viable in this system.

Attack 14

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

- **Access: 9**. The attacker does not need to have access to the system or the app but does need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive

- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.

- **Complexity: 7**

  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.

  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.

- **Effort: 7**
  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.
  - **Human: 7**. One or two people working together would be able to implement this attack.
- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.
- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.
- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. Most email or other systems have filters that will catch a lot of phishing attempts.
- **Damage: 2**
  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 5.875

Attack 15

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information in the documentation for Covid Alert on how the one-time verification codes are created. It does however state that they are random. There is information in the documentation that the code is verified when entered [22]. Such verification appears to be a simple task of comparing the code entered to a valid code list. Thus, forging a code would be blocked by the verification of the code, and this path is not viable in this system.

Attack 16

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

Though the documentation states that a one-time code will be removed from the list of valid codes as soon as it is used [22]. Thus, this attack is not technically being modeled as a replay attack, as the code will not be used twice. This attack will be modeled as an attacker getting their hands on a one time code some other way such as bribing a patient.

- **Access: 5**. This attack does not require the attacker to have higher access to the system or the app but it does require them to have access to someone who has tested positive.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. This attack could require a computer but does not have to.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 8.5**
  - **Planning: 10**. There are very few steps that required by this attack. The attacker just has to convince someone to hand over their one time code.
  - **Human: 7**. The attacker needs a person willing to give them their one-time code. Thus it requires two people.
- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.

- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.
- **Detection: 7**. The only point at which an attacker might be detected would be if the person that they took the one-time code from informed an authority. Difficult to gauge if that would occur or not.
- **Damage: 2**
  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 6.438

Attack 17

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

In the documentation for the Covid Alert app it is noted that while typically the IP address tied to a server request is deleted if an incorrect one-time code is entered it is held for a sixty minutes. Then after eight incorrect tries within the sizty minute window the IP address receives an IP ban for sixty minutes [22]. According to an individual testing the system it is not eight but fifty incorrect attempts that result in an IP ban[72]. Though this opens the opportunity for a brute force attack there is still the issue of the one-time code being a 10-12 digit pin. If it is only numerical then that is 10^{12} or 1 trillion possibilities. Due to differences in speed of languages and computers it is difficult to give a specific amount of time that brute forcing all of those codes would require however it is safe to say it would be over an hour and less than ten hours [18]. The list of valid one-time codes is changing as users use the ones that have been handed out. Thus though the target is not a single pin that needs to be found there are changes

being made as it is being searched for. However only the specific IP of the device that sent the invalid one-time code is banned. If after every 50 tries the attacker were to request a new IP from their router or use a VPN service that allows them to change their IP this ban could be circumvented. Which is the attack model being used here.

- **Access: 10**. The attacker is only using the user interface.
- **Knowledge: 10**. A brute force attack is novice level, even with the additional work around for the IP ban.
- **Complexity: 8.5**
  - **Technology: 7**. A computer would be required to create the code to try all of the possible one-time pins.
  - **Build: 10**. A single person could create this entire attack in less than a month.
- **Effort: 10**
  - **Planning: 10** Minimal components, the steps are straight forwards this is a simple attack.
  - **Human: 10** One person is all that is required to perform this attack.
- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.
- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.
- **Detection: 1** The system does monitor for this, though the banned IP is supposed to be deleted once the ban is over it would potentially not take the administrators long to recognize that someone was attempting this.
- **Damage: 2**

  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 6.313

8.2.2 Singapore TraceTogether App

Attack 1

your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 1** For the TraceTogether centralized Bluetooth system the only way to know another user's ID would be to somehow access the list on the secure server, or to access the user's ID list. This is a severe limitation to an attacker taking advantage of this vulnerability. There is nothing in the documentation that indicates the collection of GPS data when a phone makes a BLE handshake/exchange is prevented. The contact list on the device is claimed to be "secured" without a clear definition of what that entails. It is likely at the least under the usual protections of any app data, that is other apps cannot read or write it, though a user with root access could see the information. It is not clear if this data is encrypted.
- **Knowledge: 3** The implementation would require being able to make the GPS logging code and detect that the device has logged the Bluetooth information. Gaining access to the IDs would require significant knowledge.
- **Complexity: 6**
  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The analysis can be done on a consumer computer.
  - **Build: 5** The attack requires one or two tools, the GPS logger, the access to the contact list, and the method of collecting the IDs to track. Then the program to match the ID and every GPS coordinate and map them together. One person could

do this in more than a month, or a few could do this in about a month. All of these components have been created before for other attacks (the GPS data attacks of section 3.2.3) and would need to be put together for this one.

- **Effort: 1.5**
  - **Planning: 1** This is high effort because the code has to be placed on many phones to collect all of the required information. Then the data has to be compiled.
  - **Human: 2** To effectively cover an area quite a few people would need to be involved. Once the software is on their phone they would not have to necessarily do anything out of the ordinary however just go through their usual routine.
- **Scope: 3** This would affect the group IDs could be identified for. People close to any of those involved whose devices could be accessed to get the IDs would be the ones targeted.
- **Impact: 4**
  - **Data: 4** The contact log information itself is not easy to use, but the GPS data allows the creation of a map of where someone has been and could then determine where they live or something about them from that as seen in the attacks discussed in 3.2.3.
  - **Trust: 4** This would damage trust in the system. There is the potential that a politically motivated group could use this to single out certain people. Especially if they can target those that have the virus.
- **Detection: 8** It is possible that this would not be noticed, the code that collects the info does not need to interfere with the app, just monitor it, though the amount of people involved would require quite the vow of secrecy. It does involve access to someone's phone to retrieve the IDs which would increase the risk of detection.
- **Damage: 2.5**
  - **System: 1** The code added to the groups devices would have to be removed but otherwise the system itself has not truly been touched

  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of GPS location data discussed in section 2.2.

Average: 3.625

Attack 2

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 1** For the TraceTogether centralized Bluetooth system the only way to know another user's ID would be to somehow access the list on the secure server, or to access the user's ID list. This is a severe limitation to an attacker taking advantage of this vulnerability. The contact list on the device is claimed to be "secured" without a clear definition of what that entails. It is likely at the least under the usual protections of any app data, that is other apps cannot read or write it, though a user with root access could see the information. It is not clear if this data is encrypted.

- **Knowledge: 3** A lot of the work seems graduate level to create the code to mimic the Bluetooth transmissions of the Bluetrace protocol, or you would just need to have the app on the device that is then sending it through the BLE antennas. Accessing the IDs is an issue that ups the difficulty because you likely would need root access on another user's device. Thus, this is between expert and career individual.

- **Complexity: 6.5**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The tracing of IDs can be done on a consumer computer.

  - **Build: 6** The creation of the the Bluetooth devices is something that a single person could do with enough time. Creating enough of them to cover a larger area would be time consuming, as would the actual setup of the devices in the locations.

- **Effort: 6**

  - **Planning: 5** Medium to low effort, the BLE components need to be set up and placed in the area. Retrieving the IDs from a victim's phone would require care. Though taking them off the server could remove this factor.
  - **Human: 7** One or two people could do this, setting up the BLE devices might require two people.

- **Scope: 3** This would affect the group IDs could be identified for. People close to any of those involved whose devices could be accessed to get the IDs would be the ones targeted.

- **Impact: 4**

  - **Data: 4** The attack provides information on everywhere an individual has been over an area covered by the BLE devices. Though this might not be able to be used to determine where they live there remains information that could be used to identify them. This was discussed in section 3.2.3.
  - **Trust: 4** Due to the promises of not being able to track people with the app this would make people wary about downloading the app.

- **Detection: 8** It is possible that this attack would not be noticed, the only time it could be would be when an attacker tried to take the ID if the attacker was trying to get it from the victim's phone.

- **Damage: 2**

  - **System: 0** Nothing directly damaging to the system has been done.
  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of location data discussed in section 2.2.

Average: 4.188

Attack 3

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

This app is created using the BlueTrace protocol. This means that the temporary ID that is broadcast is randomly created through cryptographic processes. The temp IDs are created by using AES encryption on a random user ID and start/end time. Even if AES was breakable there is nothing to link the user to the ID unless someone had access to the server information. Assuming that the health authority is not the threat this attack is not viable here.

Attack 4 and 4.5

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

Due to this app using the Bluetooth centralized model of BlueTrace there is no server information for a user to take to determine the contact on their own. An attacker would have to use the app and wait for it to tell them when a contact was made. However, the Singapore app does not tell the user when the contact that may have exposed them was. Instead it just tells them they were exposed within the last 14 days. Thus there is no information for the attacker to use to determine which person they took a photo of was the one that has the virus.

Attack 5

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

- **Access: 10**. The access of a regular user is all that is required for this attack.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. The only computer required is the phone itself.
  - **Build: 10**. A single person could create this in less than a month as there are no extra components required.
- **Effort: 10**
  - **Planning: 10**. Very low effort, there are few components or steps.
  - **Human: 10**. One person could perform this alone.

- **Scope: 1**. Could effect one person that the attacker knows. Then perhaps be used again. Long term the attacker could use this to effect a few more people.
- **Impact: 4.5**
  - **Data: 7**. The attacker gains knowledge of whether one person has the virus. This is protected health data.
  - **Trust: 2**. Though the scope is small, this is the kind of attack that could lower the trust of some vulnerable communities or people that are unlikely to trust the system to begin with.
- **Detection: 10**. It would be almost impossible for the system to catch that this has occurred.
- **Damage: 3.5**
  - **System: 0**. The attack does not touch the system.
  - **User: 7**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 7.375

Attack 6

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy

In the documentation for the TraceTogether app there is no information about the type of security being used on the data in transit between the server and the app. As it is unclear this analysis will be assuming that the communication is being made using http not https. Thus an eavesdropping attacker can listen in on the communication.

- **Access: 7**. The attacker would need to be using a tool that provides them with information of what is being sent over the WiFi network. Many are available.
- **Knowledge: 9**. Almost novice level, this does not require any field specific knowledge, but the attacker would need to be familiar with the tool they are using. There is a lot of software available that allows attackers to eavesdrop on a WiFi network.

- **Complexity: 8.5**
  - **Technology: 7**. The monitoring of the network would require a computer.
  - **Build: 10**. A single person could create this attack in less than a month.
- **Effort: 6**
  - **Planning: 5**. The attacker has to be in the right place with their computer or a device connected to the same WiFi network to be able to see the information being passed from the server. The other component is determining who exactly the message was sent to.
  - **Human: 7**. One or two people working together could perform this exploit.
- **Scope: 5**. The attacker could target a small group of people. The targets do not have to be people that the attacker would know. Anyone on the same WiFi network could receive an exposure notification at any time.
- **Impact: 4**
  - **Data: 4**. The attacker gains the information that a user on the WiFi came into contact with someone with the virus. The attacker would need to determine who it was on the WiFi network that received the notification.
  - **Trust: 4**. As the authority assures the users that the system is private the ability for an attacker to learn that someone has received a contact message would lower trust in the system.
- **Detection: 9**. It is possible that this would not be noticed, the attack does not need to interfere with the app, just monitor the communications. The attacker physically being there would increase the risk of detection.
- **Damage: 2**
  - **System: 0**. The attack does not touch the system.
  - **User: 4**. The data that the attacker has would be private health information about the user but the attacker has only learned that the user might have the virus and are being informed to quarantine and get tested.

Average: 6.313

Attack 7

In the documentation for the TraceTogether app there is no information about the type of security being used on the data in transit between the server and the app. As it is unclear this analysis will be assuming that the communication is being made using http not https. Thus an eavesdropping attacker can listen in on the communication. In the centralized system used by the TraceTogether app the user needs to upload their contact list to the server in the case of a positive diagnosis. This is the communication that the eavesdropper would be listening for.

- **Access: 7**. The attacker would need to be using a tool that provides them with information of what is being sent over the WiFi network. Many are available. They would also need to be in an area the WiFi network can reach, be physically present.
- **Knowledge: 9**. Almost novice level, this does not require any field specific knowledge, but the attacker would need to be familiar with the tool they are using. There is a lot of software available that allows attackers to eavesdrop on a WiFi network.
- **Complexity: 8.5**
  - **Technology: 7**. This attack only requires a computer.
  - **Build: 10**. A single person could create this attack in less than a month.
- **Effort: 6**
  - **Planning: 5**. The attacker needs to be in the right place at the right time to see the information being sent to that server. If they were able to find a clinic or somewhere that patients might receive the results of tests then they might increase the chances of catching this exchange. Such a thing would require a bit of effort to work out.
  - **Human: 7**. One or two people working together could perform this exploit.
- **Scope: 4**. The attacker is limited by having to be somewhere that an individual would be uploading this information from. Finding a location that a larger group of people would do this within would be difficult. Thus the scope is limited.
- **Impact: 4**
  - **Data: 4**. The attacker gains the information that a user on the WiFi has the virus. The attacker would need to determine who it was on the WiFi network that received the notification.

  - **Trust: 4**. As the authority assures the users that the system is private the ability for an attacker to learn that someone has the virus would lower trust in the system.

- **Detection: 9**. It is possible that this would not be noticed, the attack does not need to interfere with the app, just monitor the communications. The attacker physically being there would increase the risk of detection.

- **Damage: 3**

  - **System: 0**. The attack does not touch the system.

  - **User: 6**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.313

Attack 8

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

- **Access: 10**. The attack requires no system access, it only requires that something be placed in a physical location.

- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.

- **Complexity: 9**

  - **Technology: 8**. This is possible with a device that would not have to be a computer, however could require a computer to build. Various devices can already prevent Bluetooth signals like microwave interference.

  - **Build: 10**. One person could create this attack in less than a month.

- **Effort: 7.5**

  - **Planning: 7**. To cover a large area an attacker would need multiple devices, as well as the placement of the devices to be discreet.

  - **Human: 8**. One or two people working together could create all of the device and place them in an area.

- **Scope: 6**. While not able to target an entire demographic a large area could be covered where anyone within would not receive any contact logs on their app.

- **Impact: 1.5**

  - **Data: 1**. The attack does introduce false data on the device, in that the device thinks that there are no devices close enough for a contact despite this not being the actual case. However, the data goes no farther than that into the system.

  - **Trust: 2**. This would cause people to ask why even use the app if the signals can be blocked. Temporarily lowering their trust in the system.

- **Detection: 4**. This attack would interfere with other Bluetooth devices in the same area, and potentially other signals such as WiFi. That makes this attack somewhat noticeable to anyone in the area.

- **Damage: 2**

  - **System: 0**. The attack does not touch the system.

  - **User: 4**. People in that area could have come into contact with the virus and not received the exposure notifications. A single test once the attack was discovered would let them know they are in the clear.

Average: 6.25

Attacks 9, 10, 11

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

The TraceTogether app is based on the centralized Bluetooth system. In this system the contact logs of sick individuals are uploaded processed by the server to determine the individuals at risk and then those people are informed of that risk. In this system unless an individual were to have access to the secured server there is no way for them to determine an ID belonging to a sick individual. Even if the attacker could determine the ID of a sick individual and inject it into the victim's contact list the system would never look at it. As only the contacts of sick individuals are processed. The injected information would just be deleted after 14 days.

Attack 12

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

For the TraceTogether system a user receives the one-time code used to upload their data via an SMS message. The user provides their phone number to at the testing facility. Thus, this attack would actually be the attacker asking for the one-time code and if they receive the code then they know that the user tested positive for COVID-19.

- **Access: 9**. The attacker does not need to have access to the system or the app but do need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive
- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.
- **Complexity: 7**
  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.
  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.
- **Effort: 7**

  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.
  - **Human: 7**. One or two people working together would be able to implement this attack.

- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.

- **Impact: 7**

  - **Data: 10**. The attacker would have the health information of the victims
  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.

- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. Most email or other systems have filters that will catch a lot of phishing attempts.

- **Damage: 3.5**

  - **System: 0**. The attack does touch the system.
  - **User: 7**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.438

Attack 13

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information about the process used to create the one-time code for the TraceTogether system. Neither is there any information about how the code is validated. Due to this it is not possible to assess in the manner of this thesis how vulnerable the system is to this type of attack.

Attack 14

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

For the TraceTogether system a user receives the one-time code used to upload their data via an SMS message. The user provides their phone number to at the testing facility. Thus, this attack would actually be the attacker asking for the one-time code from the user and pretending to be the place that the user has to enter the code into.

- **Access: 9**. The attacker does not need to have access to the system or the app but does need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive
- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.
- **Complexity: 7**
  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.
  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.
- **Effort: 7**
  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.

  - **Human: 7**. One or two people working together would be able to implement this attack.

- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.

- **Impact: 4**

  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.

- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. SMS has some protections against spam.

- **Damage: 2**

  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 5.875

Attack 15

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information about the process used to create the one-time code for the TraceTogether system. Neither is there any information about how the code is validated. Due to this it is not possible to assess in the manner of this thesis how vulnerable the system is to this type of attack.

Attack 16

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

- **Access: 5**. This attack does not require the attacker to have higher access to the system or the app but it does require them to have access to someone who has tested positive.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. This attack could require a computer but does not have to.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 8.5**
  - **Planning: 10**. There are very few steps that required by this attack. The attacker just has to convince someone to hand over their one time code.
  - **Human: 7**. The attacker needs a person willing to give them their one-time code. Thus it requires two people.
- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.
- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.
- **Detection: 7**. The only point at which an attacker might be detected would be if the person that they took the one-time code from informed an authority. Difficult to gauge if that would occur or not.
- **Damage: 2**

  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 6.438

Attack 17

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

There is no information in the TraceTogether documentation about the size of the one-time code required for the user to upload their contact list to the server. Nor whether there are any protections on the server against this type of attack.

- **Access: 10**. The attacker is only using the user interface.
- **Knowledge: 10**. A brute force attack is novice level, even with the additional work around for the IP ban.
- **Complexity: 8.5**
  - **Technology: 7**. A computer would be required to create the code to try all of the possible one-time pins.
  - **Build: 10**. A single person could create this entire attack in less than a month.
- **Effort: 10**
  - **Planning: 10** Minimal components, the steps are straight forwards this is a simple attack.
  - **Human: 10** One person is all that is required to perform this attack.
- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.
- **Impact: 4**

  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.

- **Detection: 7** The system does not necessarily monitor for this, though IP addresses are logged by any server when a communication is made. If an administrator was looking in those logs they would see this attack occurring.

- **Damage: 2**

  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 7.063

8.2.3 India Aarogyu Setu

At the time of writing this the only information available about the identifiers that the Aarogya Setu app uses for its Bluetooth broadcasts is that "All interactions between two devices that have the Aarogya Setu app installed, and between the device and the Aarogya Setu server are done using DiD only" [125]. DiD is the Aarogyu Setu term for the anonymous user ID. Since there is nothing that specifies that the ID is rotated or altered it will be assumed that a single ID is continuously used.

Attack 1

1. Capture your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 3** The documentation says the contact list itself is encrypted with AES using the system keychains which indicates they can be read if the device is rooted. The app does try to check if the device is rooted but this can be circumvented which is admitted by the

creators [125]. Also because the broadcast ID of the person does not change that means that someone only has to access their own devices contact list to gather the ID and then follow it access the ID on the victim's device.

- **Knowledge: 4** The implementation would require being able to make the GPS logging code and detect that the device has logged the Bluetooth information. Gaining access to the IDs would require significant knowledge. However, in this case the device is already collecting GPS data every 30 seconds, and if someone can access their own contact list they can access the GPS information as well, lowering the knowledge level required.

- **Complexity: 6**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The analysis can be done on a consumer computer.

  - **Build: 5** The attack requires one or two tools, the GPS logger, the access to the contact list, and the method of collecting the IDs to track. Then the program to match the ID and every GPS coordinate and map them together. One person could do this in more than a month, or a few could do this in about a month. All of these components have been created before for other attacks (the GPS data attacks of section 3.2.3) and would need to be put together for this one.

- **Effort: 1.5**

  - **Planning: 1** This is high effort because the code has to be placed on many phones to collect all of the required information. Then the data has to be compiled.

  - **Human: 2** To effectively cover an area quite a few people would need to be involved. Once the software is on their phone they would not have to necessarily do anything out of the ordinary however just go through their usual routine.

- **Scope: 4** This would affect the group IDs could be identified for. People close to any of those involved who can recognize someone when they are together and then use their own contact list to find the victim's ID to be traced. This allows a greater number of people to be targeted.

- **Impact: 4.5**

  - **Data: 5** The contact log information itself is not easy to use, but the GPS data allows the creation of a map of where someone has been and could then determine where they live or something about them from that as seen in the attacks discussed in 3.2.3. Having a constant ID increases the ease of creating such a map.
  - **Trust: 4** This would damage trust in the system. There is the potential that a politically motivated group could use this to single out certain people. Especially if they can target those that have the virus.

- **Detection: 8** It is possible that this would not be noticed, the code that collects the info does not need to interfere with the app, just monitor it, though the amount of people involved would require quite the vow of secrecy. Though the app does try to detect if the phone is rooted, it was said that this can be circumvented [125].

- **Damage: 2.5**

  - **System: 1** The code added to the groups devices would have to be removed but otherwise the system itself has not truly been touched
  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of GPS location data discussed in section 2.2.

Average: 4.188

Attack 2

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 3** The documentation says the contact list itself is encrypted with AES using the system keychains which indicates they can be read if the device is rooted. The app does try to check if the device is rooted but this can be circumvented which is admitted by the creators [125]. Also because the broadcast ID of the person does not change that means that someone only has to access their own devices contact list to gather the ID and then follow it access the ID on the victim's device.

- **Knowledge: 4** A lot of the work seems graduate level to create the code to mimic the Bluetooth transmissions of the Bluetrace protocol, or you would just need to have the app on the device that is then sending it through the BLE antennas. Accessing the IDs is an issue that ups the difficulty because you would need root access on your device. Thus, this is between a career individual.

- **Complexity: 6.5**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The tracing of IDs can be done on a consumer computer.

  - **Build: 6** The creation of the the Bluetooth devices is something that a single person could do with enough time. Creating enough of them to cover a larger area would be time consuming, as would the actual setup of the devices in the locations.

- **Effort: 6**

  - **Planning: 5** Medium to low effort, the BLE components need to be set up and placed in the area. Retrieving the IDs from a victim's phone would require care. Though taking them off the server could remove this factor.

  - **Human: 7** One or two people could do this, setting up the BLE devices might require two people.

- **Scope: 4** This would affect the group IDs could be identified for. People close to any of those involved who can recognize someone when they are together and then use their own contact list to find the victim's ID to be traced. This allows a greater number of people to be targeted.

- **Impact: 4**

  - **Data: 4** The attack provides information on everywhere an individual has been over an area covered by the BLE devices. Though this might not be able to be used to determine where they live there remains information that could be used to identify them. This was discussed in section 3.2.3.

  - **Trust: 4** Due to the promises of not being able to track people with the app this would make people wary about downloading the app.

- **Detection: 8** It is possible that this would not be noticed, the code that collects the info does not need to interfere with the app, just monitor it, though the amount of people involved would require quite the vow of secrecy. Though the app does try to detect if the phone is rooted, it was said that this can be circumvented [125].
- **Damage: 2**
  - **System: 0** Nothing directly damaging to the system has been done.
  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of location data discussed in section 2.2.

Average: 4.688

Attack 3

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

The only thing known about the DiD is what is claimed. It is created to be an anonymous, randomized unique device identity number. If we assume this to be the case (which we are assuming) then there is nothing to break as reversing the process will give nothing that the attack could use.

Attack 4

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

The Aarogyu Setu app is a centralized Bluetooth system similar to that of Singapore's TraceTogether app. However the Aarogyu Setu app has added functionality that Tracetogether does not. One of the added features is that the app informs the user of the health status of people they have previously come into contact with. The app informs a user directly what the status of someone they were near at a specific time in the previous days is. The system uses green, yellow, or red to indicate low risk, potential for exposure, and sick. Thus this attack can

be implemented using a camera and matching to the contact made at the same time as the time stamp without building any other tools.

- **Access: 10** The information given to the user through the app means that an attacker only needs regular user access to the system
- **Knowledge: 10** This does not require any field specific knowledge
- **Complexity: 9.5**
  - **Technology: 9** An exploit would require the additional component of a camera, though modern phones have cameras in them.
  - **Build: 10** One person can easily build this attack as all that is required is setting up the camera and device in the same location
- **Effort: 9**
  - **Planning: 8** Low, the only components are the camera and device with the app installed, then identifying the individual
  - **Human: 10** One person of this knowledge level would be able to create this and then go place it somewhere
- **Scope: 5** Affects a small group, as many people as an attacker could get to walk through a specific area.
- **Impact: 5.5**
  - **Data: 4** The attacker has gained information about who is actually sick, which is private health information, but they need to identify the people in the photo.
  - **Trust: 7** This could damage trust as the attack provides an attacker with the photo of someone who is sick, and it is very easy for an attacker to implement.
- **Detection: 8** It is possible that this attack would not be noticed, the only time it could be would be if someone noticed the camera or if an attacker did something public to determine someone's identity.
- **Damage: 3.5**
  - **System: 0** Nothing directly damaging to the system has been done.

  - **User: 7** Someone has now had their privacy about health data stolen from them, which is private data that has many protections for a reason see section 2.2 for examples of why this data is protected.

Average: 7.563

Attack 4.5

- 4.5 Everything is the same as attack 4 except the attack is a company/organization that already has security cameras in their facility and is just adding the BLE receivers to the building

- **Access: 10** The information given to the user through the app means that an attacker only needs regular user access to the system
- **Knowledge: 10** This does not require any field specific knowledge
- **Complexity: 9**
  - **Technology: 8** An exploit such as this would require a computer to setup the BLE devices. It certainly would require one to operate the security system.
  - Build: 10 One person can easily build this attack as all that is required is setting up the BLE devices and using the security cameras
- **Effort: 9**
  - **Planning: 8** The exploit would require the creation of many BLE devices, since they are all the same however that is not a limiting difficulty. Matching the
  - **Human: 10** one person could likely create this and then set it up within the building
- **Scope: 6** This wouldn't be a whole demographic, but would affect a large amount of people that are in that building or on company campus
- **Impact: 8**
  - **Data: 8**. They can use their own personnel files which likely have a photo for building identification purposes to identify the person, and it is even easier for them to determine that person's status with the app telling them.

  - **Trust: 8** The issue is that if they are taking this data in this manner then while people may not find out about it for a long time when they do it will be a big hit to the trust in a future system like this. Especially because they are just using the system as is

- **Detection: 10** The system has little to no way of detecting this. Even the victims would likely not notice anything as the security cameras were already there.

- **Damage: 5**

  - **System: 0** They didn't touch the system

  - **User: 10** They would have this information for a long time, they could use it in unexpected and interesting ways that could effect people for a long time without them even knowing it.

Average: 8.375

Attack 5

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

- **Access: 10**. The access of a regular user is all that is required for this attack.

- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.

- **Complexity: 10**

  - **Technology: 10**. The only computer required is the phone itself.

  - **Build: 10**. A single person could create this in less than a month as there are no extra components required.

- **Effort: 10**

  - **Planning: 10**. Very low effort, there are few components or steps.

  - **Human: 10**. One person could perform this alone.

- **Scope: 5**. The app's functionality allowing users to see the status of a contact made at a specific time the attacker does not have to turn the device on and off only when around a specific person. The attacker would just have to make note of the time they were alone with their target. Thus, the scope of the attack is larger as an attacker could perform this to more than one person at a time. They could even use it against someone they are not close to.

- **Impact: 7**

  - **Data: 7**. The attacker gains knowledge of whether someone has the virus. This is protected health data.

  - **Trust: 7**. The scope is large for such a simple attack. Combined with the ease of performing it with the functionality of this app this would damage the trust in the system. This is a users health information being broadcast to anyone that was near enough to them.

- **Detection: 10**. It would be almost impossible for the system to catch that this has occurred.

- **Damage: 3**

  - **System: 0**. The attack does not touch the system.

  - **User: 6**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 8.125

Attack 6 and 7

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy

7. Access a WiFi network → Capture messages between app and server → See IDs being sent from user to server indicating Covid+ status → Determine that person has the virus → Break privacy

In the documentation for the Aarogyu Setu app it is detailed that the communications with the server are encrypted and secured [125]. Breaking HTTPS or Secure TLS is beyond the scope of this thesis. Thus it is not possible for an attacker to eavesdrop on the communications as required by both of these avenues of vulnerability. It is unknown if the servers that the app communicate with are different depending on the kind of communication. Such a thing could open up the opportunity for this type of attack.

Attack 8

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

- **Access: 10**. The attack requires no system access, it only requires that something be placed in a physical location.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 9**
  - **Technology: 8**. This is possible with a device that would not have to be a computer, however could require a computer to build. Various devices can already prevent Bluetooth signals like microwave interference.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 7.5**
  - **Planning: 7**. To cover a large area an attacker would need multiple devices, as well as the placement of the devices to be discreet.
  - **Human: 8**. One or two people working together could create all of the device and place them in an area.
- **Scope: 6**. While not able to target an entire demographic a large area could be covered where anyone within would not receive any contact logs on their app.
- **Impact: 1.5**
  - **Data: 1**. The attack does introduce false data on the device, in that the device thinks that there are no devices close enough for a contact despite this not being the actual case. However, the data goes no farther than that into the system.

  - **Trust: 2**. This would cause people to ask why even use the app if the signals can be blocked. Temporarily lowering their trust in the system.

- **Detection: 4**. This attack would interfere with other Bluetooth devices in the same area, and potentially other signals such as WiFi. That makes this attack somewhat noticeable to anyone in the area.

- **Damage: 2**

  - **System: 0**. The attack does not touch the system.

  - **User: 4**. People in that area could have come into contact with the virus and not received the exposure notifications. A single test once the attack was discovered would let them know they are in the clear.

Average: 6.25

Attack 9, 10, 11

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

The Aarogyu Setu app is based on the centralized Bluetooth system. In this system the contact logs of sick individuals are uploaded processed by the server to determine the individuals at risk and then those people are informed of that risk. In this system unless an individual were to have access to the secured server there is no way for them to determine an ID belonging to a sick individual. Even if the attacker could determine the ID of a sick individual and inject it into the victim's contact list the system would never look at it. As only the contacts of sick individuals are processed. The injected information would just be deleted after 14 days.

Attack 12-17

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

In the Aarogyu Setu app system the user does not authenticate or consent to the upload of information. Once the user is using the system they have given their consent. As such the information from their phone is uploaded without them performing an action or entering a code. The health authority server that holds the test results passes information to the back-end server of the app which flags the user and pulls the data from their phone onto the server when the phone next connects. Thus, due to the nature of this system attacks 12-17 are not avenues that an attacker could use to try and attack the system.

8.2.4 Iceland Rakning C-19

Attacks 1-5

1. Capture your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

In this system there is nothing being publicly broadcast. Since there are no IDs being broadcast there is nothing for an attacker to receive and trace as a victim moves. All the system does is store the GPS location data of the user periodically. Thus none of these five potential vulnerabilities apply to this system of tracing.

Attack 6 and 7

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy

7. Access a WiFi network → Capture messages between app and server → See IDs being sent from user to server indicating Covid+ status → Determine that person has the virus → Break privacy

In the documentation for the Rakning C-19 app it is detailed that the communications with the server are encrypted and secured [123]. Breaking HTTPS or Secure TLS is beyond the

scope of this thesis. Thus it will be considered not possible for an attacker to eavesdrop on the communications as required by both of these avenues of vulnerability. It is unknown if the servers that the app communicate with are different depending on the purpose of communication. Such a thing could open up the opportunity for this type of attack.

Attack 8

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

Due to the nature of the Rakning C-19 system there is no Bluetooth signal to block. The system only stores the GPS location data of the device periodically. Thus, there is no avenue of attack by blocking the signal with this model of contact tracing.

Attacks 9, 10, 11

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

The Rakning C-19 system is GPS based. Therefore this system does not compare contacts of individuals at all. To perform this attack an attacker would have to access all of the GPS logs of the app and add in a new GPS log while removing the real one that was recorded over the same time period as the false one. However due to the nature of the system there is no practical way for an attacker to know where a COVID-19 positive patient was at a specific time to create the false GPS trail around. Thus, these attacks are not considered possible avenues of vulnerability for this app.

Attack 12-17

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

For the Rakning C-19 system the documentation says that the location information is uploaded to the server if the contact tracing team believes that it is required for contact tracing [123]. At this time it is unclear what exactly results in a user being flagged for data upload. Thus, due to the nature of this system attacks 12-14 are not avenues that an attacker could use to try and attack the system.

8.2.5 France TousAntiCovid

Attack 1

1. Capture your own GPS location when recording a contact → Pool contact lists and locations with other users into master list → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 1** For the TousAntiCovid centralized Bluetooth system the only way to know another user's ID would be to somehow access the list on the secure server, or to access the user's ID list. This is a severe limitation to an attacker taking advantage of this vulnerability. There is nothing in the documentation that indicates the collection of GPS data when a phone makes a BLE handshake/exchange is prevented. The contact list on the device is claimed to be "secured" without a clear definition of what that entails. It is likely at the least under the usual protections of any app data, that is other apps cannot read or write it, though a user with root access could see the information. It is not clear if this data is encrypted.

- **Knowledge: 3** The implementation would require being able to make the GPS logging code and detect that the device has logged the Bluetooth information. Gaining access to the IDs would require significant knowledge.

- **Complexity: 6**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The analysis can be done on a consumer computer.

  - **Build: 5** The attack requires one or two tools, the GPS logger, the access to the contact list, and the method of collecting the IDs to track. Then the program to match the ID and every GPS coordinate and map them together. One person could do this in more than a month, or a few could do this in about a month. All of these components have been created before for other attacks (the GPS data attacks of section 3.2.3) and would need to be put together for this one.

- **Effort: 1.5**

  - **Planning: 1** This is high effort because the code has to be placed on many phones to collect all of the required information. Then the data has to be compiled.

  - **Human: 2** To effectively cover an area quite a few people would need to be involved. Once the software is on their phone they would not have to necessarily do anything out of the ordinary however just go through their usual routine.

- **Scope: 3** This would affect the group IDs could be identified for. People close to any of those involved whose devices could be accessed to get the IDs would be the ones targeted.

- **Impact: 4**

  - **Data: 4** The contact log information itself is not easy to use, but the GPS data allows the creation of a map of where someone has been and could then determine where they live or something about them from that as seen in the attacks discussed in 3.2.3.

  - **Trust: 4** This would damage trust in the system. There is the potential that a politically motivated group could use this to single out certain people. Especially if they can target those that have the virus.

- **Detection: 8** It is possible that this would not be noticed, the code that collects the info does not need to interfere with the app, just monitor it, though the amount of people involved would require quite the vow of secrecy. It does involve access to someone's phone to retrieve the IDs which would increase the risk of detection.

- **Damage: 2.5**

  - **System: 1** The code added to the groups devices would have to be removed but otherwise the system itself has not truly been touched

  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. This leads back to the dangers of GPS location data discussed in section 2.2.

Average: 3.625

Attack 2

2. Setup BLE antennas to pick up Bluetooth messages → Access a user's list of their own IDs → Trace a user's ID through the areas covered → Link an id to a definite location → Break privacy

- **Access: 1** For the TousAntiCovid centralized Bluetooth system the only way to know another user's ID would be to somehow access the list on the secure server, or to access the user's ID list. This is a severe limitation to an attacker taking advantage of this vulnerability. There is nothing in the documentation that indicates the collection of GPS data when a phone makes a BLE handshake/exchange is prevented. The contact list on the device is claimed to be "secured" without a clear definition of what that entails. It is likely at the least under the usual protections of any app data, that is other apps cannot read or write it, though a user with root access could see the information. It is not clear if this data is encrypted.

- **Knowledge: 3** The implementation would require being able to make the GPS logging code and detect that the device has logged the Bluetooth information. Gaining access to the IDs would require significant knowledge.

- **Complexity: 6.5**

  - **Technology: 7** The tools required for the exploit would require a standard computer to create. The tracing of IDs can be done on a consumer computer.

  - **Build: 6** The creation of the the Bluetooth devices is something that a single person could do with enough time. Creating enough of them to cover a larger area would be time consuming, as would the actual setup of the devices in the locations.

- **Effort: 6**

  - **Planning: 5** Medium to low effort, the BLE components need to be set up and placed in the area. Retrieving the IDs from a victim's phone would require care. Though taking them off the server could remove this factor.

  - **Human: 7** One or two people could do this, setting up the BLE devices might require two people.

- **Scope: 3** This would affect the group IDs could be identified for. People close to any of those involved whose devices could be accessed to get the IDs would be the ones targeted.

- **Impact: 4**

  - **Data: 4** The attack provides information on everywhere an individual has been over an area covered by the BLE devices. Though this might not be able to be used to determine where they live there remains information that could be used to identify them. This was discussed in section 3.2.3.
  - **Trust: 4** Due to the promises of not being able to track people with the app this would make people wary about downloading the app.

- **Detection: 8** It is possible that this attack would not be noticed, the only time it could be would be when an attacker tried to take the ID if the attacker was trying to get it from the victim's phone.

- **Damage: 2**

  - **System: 0** Nothing directly damaging to the system has been done.
  - **User: 4** The damage solely from this vulnerability would not be permanent, but harmful to someone's privacy as they would have been tracked. Depending on the attacker's goal there could be a large effect felt. This leads back to the dangers of location data discussed in section 2.2.

Average: 4.188

Attack 3

3. Setup BLE antennas to pick up Bluetooth messages → Capture messages being sent between user devices → Break the encryption on the transmitted ID → Link an ID to a person → Break privacy

This app is created using the ROBERT protocol. As such the Temporary IDs are created using a block cipher that uses the server key and encrypts the permanent identifier for the user's app. The specific block cipher used is left up to the developer. However, breaking any encryption scheme currently considered secure is beyond the scope of this thesis. As well even if it was breakable there is nothing to link the user to their user ID unless an attacker had access to the server information. Assuming that the health authority is not the threat this attack is not viable here

Attack 4 and 4.5

4. Setup device and camera in specific public location (doorway) → Record time and ID received → Read list of positive IDs, compare to received IDs → Connect positive ID to timestamp and photo → Link an ID to a person → Break Privacy

Due to this app using the Bluetooth centralized model of ROBERT there is no server information for a user to take to determine a contact on their own. An attacker would have to use the app and wait for it to tell them when a contact was made. However, the TousAntiCovid app does not tell the user when the contact that may have exposed them was. Instead it just tells them they were exposed within the last 14 days. Thus, there is no information for the attacker to use to determine which person they took a photo of was the one that has the virus.

Attack 5

5. Attacker only turns on their device at specific times to capture a specific person's ID → Wait to receive contact notice → Determine that person has the virus → Break Privacy

The ROBERT protocol actually has check on the server side to prevent this type of attack. When the app makes a request of its exposure status the system checks a flag for how recent the last request was and if that request returned a positive result. If the app made the request too recently then the system does not provide it a response, the same occurs if the server had returned the request with the result that it had been exposed. This means that in order to perform this attack multiple times a user would have to prove that the previous exposure had not resulted in them contracting the illness.

- **Access: 10**. The access of a regular user is all that is required for this attack.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. The only computer required is the phone itself.
  - **Build: 10**. A single person could create this in less than a month as there are no extra components required.
- **Effort: 10**

  - **Planning: 10**. Very low effort, there are few components or steps.
  - **Human: 10**. One person could perform this alone.

- **Scope: 1**. Could effect one person that the attacker knows.
- **Impact: 4.5**
  - **Data: 7**. The attacker gains knowledge of whether one person has the virus. This is protected health data.
  - **Trust: 2**. Though the scope is small, this is the kind of attack that could lower the trust of some vulnerable communities or people that are unlikely to trust the system to begin with.
- **Detection: 4**. The system does actively try to prevent this attack from occurring multiple times. However the first time it was used the system would not catch it.
- **Damage: 3**
  - **System: 0**. The attack does not touch the system.
  - **User: 6**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.563

Attack 6 and 7

6. Access a WiFi network → Capture messages between app and server → See the contact message being sent to users → Determine that person has the virus → Break privacy
7. Access a WiFi network → Capture messages between app and server → See IDs being sent from user to server indicating Covid+ status → Determine that person has the virus → Break privacy

In the documentation for the TousAntiCovid app it is detailed that the communications with the server are encrypted and secured [158]. Breaking HTTPS or Secure TLS is beyond the scope of this thesis. Thus it is not possible for an attacker to eavesdrop on the communications as required by both of these avenues of vulnerability. It is unknown if the servers that the app communicate with are different depending on the kind of communication. Such a thing could open up the opportunity for this type of attack.

Attack 8

8. Create a device to jam/flood the Bluetooth signals → Suppress contact messages (by not allowing phones to collect contacts) → Inject false information into the system

- **Access: 10**. The attack requires no system access, it only requires that something be placed in a physical location.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 9**
  - **Technology: 8**. This is possible with a device that would not have to be a computer, however could require a computer to build. Various devices can already prevent Bluetooth signals like microwave interference.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 7.5**
  - **Planning: 7**. To cover a large area an attacker would need multiple devices, as well as the placement of the devices to be discreet.
  - **Human: 8**. One or two people working together could create all of the device and place them in an area.
- **Scope: 6**. While not able to target an entire demographic a large area could be covered where anyone within would not receive any contact logs on their app.
- **Impact: 1.5**
  - **Data: 1**. The attack does introduce false data on the device, in that the device thinks that there are no devices close enough for a contact despite this not being the actual case. However, the data goes no farther than that into the system.
  - **Trust: 2**. This would cause people to ask why even use the app if the signals can be blocked. Temporarily lowering their trust in the system.
- **Detection: 4**. This attack would interfere with other Bluetooth devices in the same area, and potentially other signals such as WiFi. That makes this attack somewhat noticeable to anyone in the area.

- **Damage: 2**
  - **System: 0**. The attack does not touch the system.
  - **User: 4**. People in that area could have come into contact with the virus and not received the exposure notifications. A single test once the attack was discovered would let them know they are in the clear.

Average: 6.25

Attack 9, 10, 11

9. Learn 1 positive ID (from online list, etc) → Gain access to other device's contact list → Inject ID into device contact list → Inject false information into the system

10. Learn 1 positive ID → Setup Bluetooth spoofer to broadcast ID like a user device, and place in a busy public area → Introduce false positives as user devices log the positive contact → Inject false information into the system

11. Learn 1 positive ID → Inject fake contact using ID into your device's contact list → Upload your information to the server → Introduce false positives → Introduce false information into the system

The TousAntiCovid app is based on the centralized Bluetooth system. In this system the contact logs of sick individuals are uploaded processed by the server to determine the individuals at risk and then those people are informed of that risk. In this system unless an individual were to have access to the secured server there is no way for them to determine an ID belonging to a sick individual. Even if the attacker could determine the ID of a sick individual and inject it into the victim's contact list the system would never look at it. As only the contacts of sick individuals are processed. The injected information would just be deleted after 14 days.

Attack 12

12. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Determine that person has the virus → Break Privacy

For the TousAntiCovid system a user receives the one-time code used to upload their data either directly from their doctor, through a website that requires authentication of their identity, or the mail. Thus, this attack would actually be the attacker asking for the one-time code and if they receive the code then they know that the user tested positive for COVID-19.

- **Access: 9**. The attacker does not need to have access to the system or the app but do need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive
- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.
- **Complexity: 7**
  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.
  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.
- **Effort: 7**
  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.
  - **Human: 7**. One or two people working together would be able to implement this attack.
- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.
- **Impact: 7**
  - **Data: 10**. The attacker would have the health information of the victims

  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.

- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. Most email or other systems have filters that will catch a lot of phishing attempts.

- **Damage: 3.5**

  - **System: 0**. The attack does touch the system.
  - **User: 7**. The personal damage is the health data that has been stolen. This data cannot be altered, it is a part of someone's medical history.

Average: 6.438

Attack 13

13. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Break the encryption or security of code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information about the process used to create the one-time code for the TousAntiCovid system. Neither is there any information about how the code is validated. Due to this it is not possible to assess in the manner of this thesis how vulnerable the system is to this type of attack.

Attack 14

14. Create a fake webportal for test results, send phishing message with url, user logs into fake site, attacker steals login and accesses real portal and gets upload code/test results → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

For the TousAntiCovid system a user receives the one-time code used to upload their data either directly from their doctor, through a website that requires authentication of their identity, or the mail. Thus, this attack would actually be the attacker asking for the one-time code and if they receive the code then they know that the user tested positive for COVID-19.

- **Access: 9**. The attacker does not need to have access to the system or the app but does need access to someone's information. This would have to be a medical professional for the attacker to gain access to a test result. Or if a patient provides their one-time code this would be confirmation to the attacker that the victim tested positive
- **Knowledge: 7**. A phishing bot is a novice level tool. The creation of the fake site would require some more specific knowledge.
- **Complexity: 7**
  - **Technology: 7**. The fake website requires a computer to setup, the phishing bot requires a computer to operate.
  - **Build: 7**. With how common phishing attacks have become there is a lot of information online on how to make them. It would take a month to setup the website.
- **Effort: 7**
  - **Planning: 7**. Low effort, there are a couple of components required. Setting up the website and entering the information given into the real one would be the most intensive aspect.
  - **Human: 7**. One or two people working together would be able to implement this attack.
- **Scope: 4**. Assuming that the system only allows that a health care professional see their patient's information then the group that the attacker would have information on is small. Alternatively the attacker would be able to target a small set of the patients who fall for the phishing attack.
- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. Though phishing attacks are common enough that most people are aware they can occur people would be wary of the system after this attack.
- **Detection: 7**. In some cases phishing is detected very early, sometimes it takes longer. Most email or other systems have filters that will catch a lot of phishing attempts.

- **Damage: 2**
  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 5.875

Attack 15

15. Get an upload code from the health authorities → Break the encryption or security of the code → Forge a code → Upload false information using code → Introduce false positives → Introduce false information to the system

There is no information about the process used to create the one-time code for the TousAntiCovid system. Neither is there any information about how the code is validated. Due to this it is not possible to assess in the manner of this thesis how vulnerable the system is to this type of attack.

Attack 16

16. Get an upload code from the health authorities → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

- **Access: 5**. This attack does not require the attacker to have higher access to the system or the app but it does require them to have access to someone who has tested positive.
- **Knowledge: 10**. Novice level, this does not require any field specific knowledge.
- **Complexity: 10**
  - **Technology: 10**. This attack could require a computer but does not have to.
  - **Build: 10**. One person could create this attack in less than a month.
- **Effort: 8.5**

  - **Planning: 10**. There are very few steps that required by this attack. The attacker just has to convince someone to hand over their one time code.
  - **Human: 7**. The attacker needs a person willing to give them their one-time code. Thus it requires two people.

- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.

- **Impact: 4**

  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.

- **Detection: 7**. The only point at which an attacker might be detected would be if the person that they took the one-time code from informed an authority. Difficult to gauge if that would occur or not.

- **Damage: 2**

  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 6.438

Attack 17

17. Brute force an upload code → Replay a code → Upload false information using the code → Introduce false positives → Introduce false information to the system

There is no information in the TraceTogether documentation about the size of the one-time code required for the user to upload their contact list to the server. Nor whether there are any protections on the server against this type of attack.

- **Access: 10**. The attacker is only using the user interface.
- **Knowledge: 10**. A brute force attack is novice level, even with the additional work around for the IP ban.
- **Complexity: 8.5**
  - **Technology: 7**. A computer would be required to create the code to try all of the possible one-time pins.
  - **Build: 10**. A single person could create this entire attack in less than a month.
- **Effort: 10**
  - **Planning: 10** Minimal components, the steps are straight forwards this is a simple attack.
  - **Human: 10** One person is all that is required to perform this attack.
- **Scope: 5**. The attacker could target a large number of people by placing the device they intend to use the one-time code on somewhere with heavy foot traffic. This way a large number of people will be notified of an exposure.
- **Impact: 4**
  - **Data: 4**. The attacker has introduced false data to the system that will be deleted after 15 days
  - **Trust: 4**. This would lower trust in the system. The ease with which a person could force a group of people into quarantine would make users wary.
- **Detection: 7** The system does not necessarily monitor for this, though IP addresses are logged by any server when a communication is made. If an administrator was looking in those logs they would see this attack occurring.
- **Damage: 2**
  - **System: 0**. The attack do touch the system, but the data is automatically deleted after a time and no action is required
  - **User: 4**. The user will have been asked to quarantine for 14 days and to get a COVID test, one-time actions in response to the attack

Average: 7.063

Table 8.3: Canada Covid Alert Security Summary

Attack	Access	Knowledge	Complexity Tech	Complexity Build	Effort Planning	Effort Human	Scope	Impact Data	Impact Trust	Detection	Damage System	Damage User	Average
1	2	3	7	5	1	2	4	4	4	8	1	4	3.875
2	2	3	7	6	5	7	4	4	4	8	0	4	4.438
3	-	-	-	-	-	-	-	-	-	-	-	-	-
4	4	7	7	7	7	7	5	4	4	8	0	7	5.688
4.5	4	7	7	7	7	10	6	7	7	10	0	10	6.813
5	10	10	10	10	10	10	1	7	2	10	0	6	7.313
6	-	-	-	-	-	-	-	-	-	-	-	-	-
7	7	7	7	10	5	7	4	4	4	9	0	6	6.063
8	10	10	8	10	7	8	6	1	2	4	0	4	6.25
9	1	2	7	4	4	10	1	1	1	8	0	4	3.438
10	4	4	7	7	6	7	6	1	3	8	0	4	4.938
11	-	-	-	-	-	-	-	-	-	-	-	-	-
12	9	7	7	7	7	7	4	10	4	7	0	7	6.438
13	-	-	-	-	-	-	-	-	-	-	-	-	-
14	9	7	7	7	7	7	4	4	4	7	0	4	5.875
15	-	-	-	-	-	-	-	-	-	-	-	-	-
16	5	10	10	10	10	7	5	4	4	7	0	4	6.438
17	10	10	7	10	10	10	5	4	4	1	0	4	6.313

Table 8.4: Singapore TraceTogether Security Summary

Attack	Access	Knowledge	Complexity Tech	Complexity Build	Effort Planning	Effort Human	Scope	Impact Data	Impact Trust	Detection	Damage System	Damage User	Average
1	1	3	7	5	1	2	3	4	4	8	1	4	3.625
2	1	3	7	6	5	7	3	4	4	8	0	4	4.188
3	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-
4.5	-	-	-	-	-	-	-	-	-	-	-	-	-
5	10	10	10	10	10	10	1	7	2	10	0	7	7.375
6	7	9	7	10	5	7	5	4	4	9	0	4	6.313
7	7	9	7	10	5	7	4	4	4	9	0	6	6.313
8	10	10	8	10	7	8	6	1	2	4	0	4	6.25
9	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-
14	9	7	7	7	7	7	4	4	4	7	0	4	5.875
15	-	-	-	-	-	-	-	-	-	-	-	-	-
16	5	10	10	10	10	7	5	4	4	7	0	4	6.438
17	10	10	7	10	10	10	5	4	4	7	0	4	7.063

Table 8.5: India Aarogyu Setu Security Summary

Attack	Access	Knowledge	Complexity Tech	Complexity Build	Effort Planning	Effort Human	Scope	Impact Data	Impact Trust	Detection	Damage System	Damage User	Average
1	3	4	7	5	1	2	4	5	4	8	1	4	4.188
2	3	4	7	6	5	7	4	4	4	8	0	4	4.688
3	-	-	-	-	-	-	-	-	-	-	-	-	-
4	10	10	9	10	8	10	5	4	7	8	0	7	7.563
4.5	10	10	8	10	8	10	6	8	8	10	0	10	8.375
5	10	10	10	10	10	10	5	7	7	10	0	6	8.125
6	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-
8	10	10	8	10	7	8	6	1	2	4	0	4	6.25
9	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-

8

Table 8.6: France TousAntiCovid Security Summary

Attack	Access	Knowledge	Complexity Tech	Complexity Build	Effort Planning	Effort Human	Scope	Impact Data	Impact Trust	Detection	Damage System	Damage User	Average
1	1	3	7	5	1	2	3	4	4	8	1	4	3.625
2	1	3	7	6	5	7	3	4	4	8	0	4	4.188
3	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-
4.5	-	-	-	-	-	-	-	-	-	-	-	-	-
5	10	10	10	10	10	10	1	7	2	4	0	6	6.563
6	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-
8	10	10	8	10	7	8	6	1	2	4	0	4	6.25
9	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-
12	9	7	7	7	6	7	4	10	4	6	0	7	6.25
13	-	-	-	-	-	-	-	-	-	-	-	-	-
14	9	7	7	7	6	7	4	4	4	6	0	5	5.75
15	-	-	-	-	-	-	-	-	-	-	-	-	-
16	5	10	10	10	10	7	5	4	4	7	0	4	6.438
17	10	10	7	10	10	10	5	4	4	7	0	4	7.063

8.2.6 Summary of Application Vulnerability

None of the suggested attacks laid out in the attack tree as it turns out are viable against the Icelandic Rakning C-19 app. Thus, there is no table summarizing the vulnerability of the app as it would be blank.

8.2.7 Vulnerability Ranking of Selected Applications

Now that the vulnerability of the applications has been assessed against the proposed attack vectors the applications are ranked. This ranking corresponds to how serious a vulnerability in the system is. Green indicates that nothing of serious concern was discovered. Yellow means that there are some concerning areas in the security of the system. Red indicates that critical security flaws are present in the system. As described in the 7.3 section this initial ranking is performed the same as the initial ranking performed when creating the Common Vulnerability Scoring System (CVSS) [47]. The ranking is done by the reviewer based on the assessment against the rubric. Then the rankings are used to determine the define the acceptable numeric ranges for each severity level.

Initial Ranking of Contact Tracing Applications

- Canada Covid Alert - Yellow
- Singapore TraceTogether - Yellow
- Iceland Rakning C-19 - Green
- India Aarogya Setu - Red
- France TousAntiCovid - Yellow

Iceland's Rakning C-19 application is ranked as green because of the lack of vulnerability. When the proposed attack tree was run against the system of the Rakning C-19 application none of the proposed potential vulnerabilities were found to be a viable option against the system. This is due to the nature of GPS logging. The system is quite secure against these types of attacks. Thus, the application receives a ranking of Green.

India's Aarogya Setu application is ranked as red because of the nature of the vulnerabilities found. When the proposed attack tree was run against the system of the Aarogya Setu application two of the vulnerabilities received a totalled average of higher than 8. The functionalities within the system leave opportunities for attackers to take advantage and maliciously target users. Allowing the users to see how many individuals near them currently, in a specific area, or that have been near them have symptoms or the virus is a dangerous amount of information to be publishing.

Canada's Covid Alert application is ranked as yellow because while many of the proposed vulnerabilities have the potential to be exploited within the system the highest has a totalled average of 7.3. This is vulnerability is concerning but not necessarily a critical issue with the system. Thus, the application receives a ranking of Yellow.

Singapore's TraceTogether application is ranked as yellow because similarly to the Covid Alert app while there are vulnerabilities that have the potential to be exploited and two with average totals of 7 these are only concerning. The application has some areas that are of concern though not necessarily a critical vulnerability in the system. Thus, the application receives a ranking of Yellow.

France's TousAntiCovid application is ranked as yellow because while many of the proposed vulnerabilities have the potential to be exploited within the system the highest has a totalled average of 7.06. This is vulnerability is concerning but not necessarily a critical issue with the system. Thus, the application receives a ranking of Yellow.

Creating Ranges for Each Severity Level required reviewing the results of the vulnerability assessment and determining the method best used to differentiate between the rankings. It was decided that while the attack tree is thorough there is the potential that future vulnerabilities in the systems will be proposed. Thus, the attack tree here cannot be considered to cover the entire breadth of the vulnerability field. Due to this consideration looking at the number of vulnerabilities a system has is not as useful a metric as looking at the severity of any one vulnerability that the system has. In this way, if a vulnerability were to be found that was critical the security ranking of an app would drop into the red regardless of where it was before or whether there are any other vulnerabilities in the system.

This creates a ranking system based on the highest average total of a vulnerability proposed against the system and scored using the rubric of this thesis. It is as follows:

- Green: 0 - 3.9
- Yellow: 4 - 7.9
- Red: 8 - 10

8.3 Summary of Assessment of Contact Tracing Applications

The assessments of privacy and security were performed and used to determine the ranges for the severity level metrics. This is detailed in the previous pages of this chapter. Here is a summary of the results. Table 8.7 contains the numerical ranges for the privacy rankings of the applications. These numerical values are determined by comparing the contact tracing system to the privacy principles laid out in section 7.2.2. Such a review was performed in section 8.1. Table 8.8 contains the numerical ranges for the security rankings of the applications. These numerical values are determined for each application by assessing how severe a vulnerability is based on the rubric of section 7.3.1. Such assessments were performed for each application and all proposed attacks in section 8.2. A summary of the results of the privacy and security reviews of the 5 selected application is in table 8.9.

Table 8.7: Contact Tracing Application Privacy Scoring

Ranking	Privacy Score
Good	8 - 10
Medium	5 - 7.5
Low	0 - 4.5

Table 8.8: Contact Tracing Application Security Scoring

Ranking	Highest Severity of a Vulnerability
Good	0 - 3.9
Medium	4 - 7.9
Low	8 - 10

Table 8.9: Contact Tracing Application Security and Privacy Scoring

Ranking	Privacy Score	Vulnerability Score
Covid Alert	8.5	7.313
TraceTogether	5.5	7.375
Rakning C-19	6.5	0
Aarogya Setu	4	8.375
TousAntiCovid	5.5	7.063

Chapter 9

Discussion, Future Work, and Conclusions

Contact tracing is a topic that within a single year went from completely unknown to the general public to being regularly discussed on the nightly news. Contact tracing along with many other things has been pushed into the consciousness of the public due to the circumstances of a global pandemic. Many of those other things will slowly fade back out of the public eye once a vaccine is created and enough of the populace is inoculated to create herd immunity. That does not mean that contact tracing will not continue. Like disinfection procedures and personal protective equipment, contact tracing is one of the modern tools the medical field uses to crack down on virus spread. It is the means of catching the virus early. When there is an outbreak in the future of another virus it will be used again to try and prevent that virus from becoming a global pandemic. And whether contact tracing is used in a single area, an entire country, or the whole world it should be private and secure.

Through this thesis, we have attempted to create a tool that can be used to hold these systems to a standard of practice. A method to look at them and determine whether the good they can do through their function is belittled by the harm they could do through their functioning. By holding them under the microscope the trust that is necessary in the public for contact tracing to work can be created.

There are many potential ways that a system can be used that is unintended by the creator. Just as there are many ways that a creator can use a system that is unintended by those that permitted them to create it. What can seem to be a small amount of data over time can become a vast database that could be used to create powerful predictive models. Knowing how small privacy concessions today will impact the future is a next to impossible endeavour. This is why it is key that we do not compromise on the privacy of our systems.

9.1 Discussion

This assessment method is the first in creating a method to assess the security and privacy of digital contact tracing. The CVSS was used as the basis for the methodology. The development of the CVSS did not occur in a singular fashion it has released new versions as through experience of applying the system reveals limitations in the current version. CVSS version 3 has many differences in scoring from version 2. This assessment is also empirically derived and it is likely that as it is used there will be areas of possible improvement to it that will be discovered.

Though the attack tree has many novel attacks, like any other it cannot be proven to be complete. Other attacks could exist that were not laid out in this thesis. For the purpose of creating a security assessment methodology, this limitation was addressed in how the security level was determined based on the severity of a single vulnerability against the system. Future vulnerabilities discovered should be scored using the created method to determine if an app's security is impacted by the discovery of the new vulnerability.

9.2 Future Work

This work like many before it is not intended to be written in stone. Instead, it is intended as a step towards a method of assessing the many digital contact tracing systems that are being released. Improvements can be made to expand upon this work and create a system to assess the security and privacy of this type of software. The CVSS is currently on version 3.1, and Rome was not made in a day.

- **Expand the Privacy Review:** one area of future work would be to get a multilingual team of privacy reviewers to monitor the documentation of the apps being developed and deployed and keep an up to date publicly available review of their privacy.
- **Expand the Security Review:** a major undertaking would be to take the security review further. A larger number of apps being reviewed through the lens of a larger number of attacks would improve our understanding of the shortcomings of these systems.
  - **Comparing more apps:** a review of the state of security across the entire or a statistically significant amount of the field of these apps could be performed. By doing this information about the state of security in the entire ecosystem could be

reviewed. A ranking of the apps could be done to see how they compare to each other. Such a thing could be used to guide policy as countries look to adopt different systems.

  - **Comparing more attacks:** expanding the attack tree would improve the security review. The 18 vulnerability paths are not the only possible ways that these systems could be attacked. Potential new vulnerabilities could be presented against the systems and marked along with others to determine how severe they are. This would provide more information to developers on what they need to protect the system from.
  - **Diving Deeper Into the Apps:** if the apps were tested against the vulnerabilities further by probing their implementation that would assist developers in improving their security. As well as policymakers in the decisions they make about what apps to implement. This would by no means require, nor would it be recommended to, actually build an exploit to run against the system. This would be a further analysis of the implementation to see where the vulnerabilities could be found.

- **Fine Tune the Score Calculations:** jumping off from the expansion of both the reviews, when the reviews are expanded it would create more information to guide the equations used for score calculations. There are different ways that the scores could be further fine-tuned. This could involve a weighted average, which would require coefficients to be carefully determined. There is also the potential to multiply some metrics together or otherwise have them affect one another. For instance, in the vulnerability score scope and user damage could relate to each other as the number of people that have to deal with the damage could be viewed as making the vulnerability worse. Or the knowledge level could impact the build complexity as it affects who could be designing the system, complex for a novice might not be the same as complex for an expert.

- **Assess Breadth and Depth:** The privacy assessment metric currently uses the breadth of the privacy principles. That is, how the app does along all of them determines how private the app is. Using the depth approach a serious enough breach of any one principle could determine the privacy of the app. Oppositely in the security metric, a serious enough vulnerability results in the app being considered insecure rather than many less concerning vulnerabilities. The choices to perform the assessment in this manner are that the privacy principles did not contain any information on what would be a serious breach

of each principle. Thus, creating a level of privacy breach for each principle could be a future endeavor. The security reasoning was that the attack tree developed in this work is thorough but more vulnerabilities could exist. Thus, considering it as a whole was less important than determining how vulnerable the system was to any one path. However, a secondary method of viewing the system by looking at how many avenues of attack there are against a system could lead to valuable insights into the security of the system.

9.3 Final Remarks

In the middle of a health crisis with people's lives at risk and the potential for the exponential spread of the virus, solutions can seem necessary and worth the risk. That does not mean they are. It is important that even in times of crisis the principles that we claim to hold still matter. We do not create principles for the times when decisions are easy, but when they are difficult.

In this work, five apps have been reviewed and compared directly from a field of over fifty-five representing different countries. There are more systems out there than that and they should all be held to some standards of privacy and security. This work attempts to build a tool to provide the public with a meaningful and systematic way of reviewing the apps that are being touted to protect them. The world of innovation does not take steps backward. And privacy is not easily repaired once it is broken. Waiting until after the pandemic to determine what should have been accepted is to have waited too long. By making a metric of comparing contact tracing apps now we take a step to giving the public the tools they need to protect their privacy. An important tool in safeguarding the public in the digital age.

Bibliography

[1] Elliot Alderson. Aarogya setu: The story of a failure. *Medium*, May 2020. Available at https://medium.com/@fs0c131y/aarogya-setu-the-story-of-a-failure-3a190a18e34.

[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. *arXiv preprint arXiv:1212.1984*, 2012.

[3] Aranja. rakning-c19-app. *GitHub*, Mar 2020. Available at https://github.com/aranja/rakning-c19-app.

[4] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps. *Proceedings. Sixth International Symposium on Wearable Computers,.*

[5] Aura. Why identity theft is a real problem. *Identity Guard*, Jul 2018.

[6] Johann Bacher, Ruth Brand, and Stefan Bender. Re-identifying register data by survey data using cluster analysis: An empirical study. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):589–607, 2002.

[7] Michael Balsamo. U.s. hate crimes rise to highest level in more than a decade, fbi report says. *Global News*, Nov 2020.

[8] Michael Barbaro and Tom Zeller. A face is exposed for aol searcher no. 4417749. *The New York Times*, Aug 2006.

[9] BBC. Coronavirus: Bahrain contact-tracing app shares data with gameshow. *BBC News*, Jun 2020.

[10] Kathleen Benitez and Bradley Malin. Evaluating re-identification risks with respect to the HIPAA privacy rule. *Journal of the American Medical Informatics Association*, 17(2):169–177, 03 2010.

[11] Kathleen Benitez and Bradley Malin. Evaluating re-identification risks with respect to the hipaa privacy rule. *Journal of the American Medical Informatics Association*, 17(2):169–177, 2010.

[12] Sean Bland. Reflections on the history of contact tracing. *O'Neill Institute*, Jul 2020.

[13] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. *Information Security Technology for Applications Lecture Notes in Computer Science*, page 31–46, 2012.

[14] Eerke Boiten, Mark Ryan, and Alan Woodward. United kingdom scientific joint statement. *googledrive*, Apr 2020. Available at https://drive.google.com/file/d/1uB4LcQHMVP-oLzIIHA9SjKj1uMd3erGu/view.

[15] Masha Borak. China wants to keep health codes after the pandemic but users aren't so sure. *South China Morning Post*, Jun 2020.

[16] Stephanie Borys and Ariel Bogle. Google and apple unite to help countries like australia fix their contact tracing apps. *ABC News*, May 2020.

[17] John S. Brownstein, Christopher A. Cassa, and Kenneth D. Mandl. No place to hide — reverse identification of patients from published maps. *New England Journal of Medicine*, 355(16):1741–1742, 2006.

[18] Marius Bughiu. How long does it take a pc to count to one trillion. *Start Debugging*, Aug 2020.

[19] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *CoRR*, abs/1605.02065, 2016.

[20] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "you might also like:" privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy*, pages 231–246, May 2011.

[21] Health Canada. Canada's exposure notification app - canada.ca. *canada.ca*, Oct 2020. Available at https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert/help.html#9.

[22] Health Canada. Covid alert: Covid-19 exposure notification application privacy assessment. *canada.ca*, Oct 2020. Available at https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert/privacy-policy/assessment.html.

[23] Yinzhi Cao, Song Li, and Erik Wijmans. (cross-)browser fingerprinting via os and hardware level features. *Proceedings 2017 Network and Distributed System Security Symposium*, 2017.

[24] Amazon Care. Amazon care: Healthcare built around you. *amazon.care*, 2019. Available at https://amazon.care/faq.

[25] Ann Cavoukian. *Privacy by Design - The 7 Foundational Principles*, May 2010.

[26] Johns Hopkins Coronavirus Resource Center. Covid-19 map. 2020. Available at https://coronavirus.jhu.edu/map.html.

[27] Chaos Computer Club. 10 requirements for the evaluation of "contact tracing" apps. *CCC.de*, Apr 2020. Available at https://www.ccc.de/en/updates/2020/contact-tracing-requirements.

[28] Samuel Cohn Professor of History and Mona O'Brien Graduate Teaching Assistant in History. Contact tracing: how physicians used it 500 years ago to control the bubonic plague. *The Conversation*, Jun 2020.

[29] Federal Trade Commission. Equifax data breach settlement. *Federal Trade Commission*, Jul 2020.

[30] Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. Health data in an open world. *CoRR*, abs/1712.05627, 2017.

[31] Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. Publication of mbs/pbs data commissioner initiated investigation report. *Australian Government Office of the Australian Information Commissioner*, 2018.

[32] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[33] Zach Diamond. We know they are listening, but what do they hear? *Medium*, Apr 2017.

[34] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM.

[35] Marie Douriez, Harish Doraiswamy, Juliana Freire, and Claudio T. Silva. Anonymizing nyc taxi data: Does it matter? *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016.

[36] Kostas Drakonakis, Panagiotis Ilia, Sotiris Ioannidis, and Jason Polakis. Please forget where I was last summer: The privacy risks of public location (meta)data. *CoRR*, abs/1901.00897, 2019.

[37] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, July 2006.

[38] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[39] Svea Ecker and Andreas Dewes. Your 'anonymous' browsing data isn't actually anonymous. *Vice*, Aug 2017.

[40] PF Edemekong and MJ. Haydel. Health insurance portability and accountability act (hipaa). 2019.

[41] Jessy Edwards. Tracking coronavirus: Should you install the coronapp? *The Bogotá Post*, Jun 2020.

[42] Khaled El Emam. Data anonymization practices in clinical research a descriptive study. *Access to Information and Privacy Division of Health Canada*, May 2006.

[43] Khaled El Emam, Fida K Dankar, Angelica Neisa, and Elizabeth Jonker. Evaluating the risk of patient re-identification from adverse drug event reports. *BMC Medical Informatics and Decision Making*, 13(1), 2013.

[44] Khaled El Emam, Fida K Dankar, Régis Vaillancourt, Tyson Roffey, and Mark Lysyk. Evaluating the risk of re-identification of patients from hospital prescription records. *The Canadian Journal of Hospital Pharmacy*, 62(4), 2009.

[45] Health Service Executive. Covid tracker app: Data protection information notice (dpin). *hse.ie*, 2020. Available at https://covidtracker.gov.ie/privacy-and-data/data-protection/#11.

[46] Jie Feng, Mingyang Zhang, Huandong Wang, Zeyu Yang, Chao Zhang, Yong Li, and Depeng Jin. Dplink: User identity linkage via deep neural network from heterogeneous mobility data. *The World Wide Web Conference on - WWW 19*, 2019.

[47] FIRST. Cvss v3.1 specification document. *FIRST*, 2015. Available at https://www.first.org/cvss/v3.1/specification-document.

[48] Enterprise Big Data Framework. A short history of big data: Big data framework. *Big Data Framework*, Mar 2019.

[49] Dan Frankowski, Dan Cosley, Shilad Sen, Loren Terveen, and John Riedl. You are what you say. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 06*, 2006.

[50] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. Evaluating the privacy risk of location-based services. In George Danezis, editor, *Financial Cryptography and Data Security*, pages 31–46, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[51] Andrea Gadotti, Florimond Houssiau, Luc Rocher, and Yves-Alexandre de Montjoye. When the signal is in the noise: The limits of diffix's sticky noise. *CoRR*, abs/1804.06752, 2018.

[52] Sebastien Gambs, Marc-Olivier Killijian, and Miguel Nunez Del Prado Cortez. De-anonymization attack on geolocated data. *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013.

[53] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, pages 34–41, New York, NY, USA, 2010. ACM.

[54] Jing Gao, Lijun Sun, and Ming Cai. Quantifying privacy vulnerability of individual mobility traces: A case study of license plate recognition data. *Transportation Research Part C: Emerging Technologies*, 104:78–94, 2019.

[55] Tedros Adhanom Ghebreyesus. Who director-general's opening remarks at the media briefing on covid-19 - 11 march 2020. *World Health Organization*, Mar 2020.

[56] Albert Gidari. Manual contact tracing has privacy issues. *Center for Internet and Society*, May 2020.

[57] Daniel Kahn Gillmor. Aclu white paper - principles for technology-assisted contact-tracing. *American Civil Liberties Union*, Apr 2020.

[58] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. *Proceedings of the 5th ACM workshop on Privacy in electronic society - WPES 06*, 2006.

[59] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. *Lecture Notes in Computer Science Pervasive Computing*, page 390–397, 2009.

[60] Google. Google health: Live your healthiest life. *health.google*, 2006. Available at https://health.google/.

[61] Google. Contact tracing cryptography specification. *Contact Tracing Cryptography Specification V1.0*, Apr 2020.

[62] Google and Apple Inc. Exposure notification bluetooth® specification. *Exposure Notification Bluetooth® Specification V1.2*, Mar 2020.

[63] Google and Apple Inc. Exposure notification cryptography specification. *Exposure Notification Cryptography Specification V1.2*, Apr 2020.

[64] Australian Government. Privacy policy for covidsafe application. *covidsafe.gov.au*, Oct 2020. Available at https://covidsafe.gov.au/privacy-policy.html.

[65] IBM. Making health smarter together: Watson health. *ibm.com*. Available at https://www.ibm.com/watson/health/resources/making-health-smarter-together/.

[66] Apple Inc. ios - health. *apple.com*, 2014. Available at https://www.apple.com/ca/ios/health/.

[67] Apple Inc. Exposure notification framework. *Apple Developer Documentation*, 2020.

[68] Expii inc. *novid.org*, 2020. Available at https://www.novid.org/.

[69] Privacy International. Colombia: Coronapp fails at public information purpose. *privacyinternational.org*, Mar 2020. Available at https://privacyinternational.org/examples/3435/colombia-coronapp-fails-public-information-purpose.

[70] Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. Identifying important places in people's lives from cellular network data. *Lecture Notes in Computer Science Pervasive Computing*, page 133–151, 2011.

[71] Victor Janmey and Peter L Elkin. Re-identification risk in hipaa de-identified datasets: The mva attack. *AMIA ... Annual Symposium proceedings*, 2018 1329-1337, 12 2018.

[72] Abhineet Jayaraj. How secure is canada's covid alert app? evaluation of android app v1.0.3. *How Secure Is Canada's COVID Alert App?*, Sep 2020.

[73] Bargav Jayaraman and David Evans. When relaxations go bad: "differentially-private" machine learning. *CoRR*, abs/1902.08874, 2019.

[74] Al Jazeera. Qatar makes covid-19 app mandatory, experts question efficiency. *Qatar — Al Jazeera*, May 2020.

[75] Lukasz Jedrzejczyk, Blaine A. Price, Arosha K. Bandara, and Bashar Nuseibeh. Know what you did last summer : risks of location data leakage in mobile and social computing. 2009.

[76] S. Ji, W. Li, M. Srivatsa, and R. Beyah. Structural data de-anonymization: Theory and practice. *IEEE/ACM Transactions on Networking*, 24(6):3523–3536, December 2016.

[77] El Emam K, Buckeridge D, Tamblyn R, Neisa A, Jonker E, and Verma A. The re-identification risk of canadians from longitudinal demographics. June 2011.

[78] Florian Kerschbaum and Ken Barker. Coronavirus statement. *Waterloo Cybersecurity and Privacy Institute*, May 2020.

[79] P. Knoche. Factual anonymity of microdata from household and person related surveys - the release of microdata for scientific purposes. *Proceedings of the International Symposium on Statistical Confidentiality*, pages 407–413, 1993.

[80] Matthijs Koot, Guido Noordende, and Laat de C. A study on the re-identifiability of dutch citizens. *Journal of Clinical Virology - J CLIN VIROL*, 01 2010.

[81] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[82] John Krumm. Inference attacks on location tracks. *Lecture Notes in Computer Science Pervasive Computing*, page 127–143, 2007.

[83] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Ashley Durham, Csaba Toth, and Bradley Malin. A practical approach to achieve private medical record linkage in light of public resources. *Journal of the American Medical Informatics Association*, 20(2):285–292, 2013.

[84] Arturs Lavrenovs and Karlis Podins. Privacy violations in riga open data public transport system. *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2016.

[85] Peter Lee. Microsoft for healthcare: new people, products, and partnerships. *microsoft.com*, 2019.

[86] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, 2007.

[87] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Dependence makes you vulnerable: Differential privacy under dependent tuples. *Proceedings 2016 Network and Distributed System Security Symposium*, 2016.

[88] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM.

[89] Natasha Lomas. Researchers spotlight the lie of 'anonymous' data. *TechCrunch*, Jul 2019.

[90] Natasha Lomas. Norway pulls its coronavirus contacts-tracing app after privacy watchdog's warning. *TechCrunch*, Jun 2020.

[91] Grigorios Loukides, Joshua C Denny, and Bradley Malin. The disclosure of diagnosis codes can breach research participants privacy. *Journal of the American Medical Informatics Association*, 17(3):322–327, 2010.

[92] Ltnadmin. Terabyte definition: The interactive glossary. *Website Builders.com*, Oct 2019.

[93] Matteo Luccio. Using contact tracing and gps to fight spread of covid-19. *GPS World*, Jun 2020.

[94] Chris Y. T. Ma, David K. Y. Yau, Nung Kwan Yip, and Nageswara S. V. Rao. Privacy vulnerability of published anonymous mobility traces. *IEEE/ACM Transactions on Networking*, 21(3):720–733, 2013.

[95] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, April 2006.

[96] Bradely Malin. Re-identification of familial database records. *AMIA Annu Symp Proc.*, page 524–528, 2006.

[97] Bradley Malin and Latanya Sweeney. Determining the identifiability of dna database entries. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pages 537–41, 02 2000.

[98] Bradley Malin and Latanya Sweeney. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics*, 37(3):179–192, 2004.

[99] Jian Mao, Wenqian Tian, Jingbo Jiang, Zhaoyuan He, Zhihong Zhou, and Jianwei Liu. Understanding structure-based social network de-anonymization techniques via empirical analysis. *EURASIP Journal on Wireless Communications and Networking*, 2018(1):279, Dec 2018.

[100] Mohamed Maouche, Sonia Ben Mokhtar, and Sara Bouchenak. Ap-attack: A novel user re-identification attack on mobility datasets. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous 2017, pages 48–57, New York, NY, USA, 2017. ACM.

[101] Lisa Maragakis. Coronavirus diagnosis: What should i expect? *Johns Hopkins Medicine*, Oct 2020. Available at https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/diagnosed-with-covid-19-what-to-expect.

[102] Clément Massart and François-Xavier Standaert. Revisiting location privacy from a side-channel analysis viewpoint. *Progress in Cryptology – AFRICACRYPT 2019 Lecture Notes in Computer Science*, page 333–351, 2019.

[103] Jonathan Mayer, Patrick Mutchler, and John C. Mitchell. Evaluating the privacy properties of telephone metadata. *Proceedings of the National Academy of Sciences*, 113(20):5536–5541, 2016.

[104] Simon Denyer Min Joo Kim. A 'travel log' of the times in south korea: Mapping the movements of coronavirus carriers. *The Washington Post*, Mar 2020.

[105] Ilya Mironov. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017.

[106] MohGovIL. Hamagen source code. *GitHub*, 2020. Available at https://github.com/MohGovIL/hamagen-react-native.

[107] Yves-Alexandre De Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3(1), 2013.

[108] Paul Mozur, Raymond Zhong, and Aaron Krolik. In coronavirus fight, china gives citizens a color code, with red flags. *The New York Times*, Mar 2020.

[109] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. Identification via location-profiling in gsm networks. *Proceedings of the 7th ACM workshop on Privacy in the electronic society - WPES 08*, 2008.

[110] Liangyuan Na, Cong Yang, Chi-Cheng Lo, Fangyuan Zhao, Yoshimi Fukuoka, and Anil Aswani. Feasibility of reidentifying individuals in large national physical activity data sets from which protected health information has been removed with use of machine learning. *JAMA Network Open*, 1(8), 2018.

[111] Arvind Narayanan and Edward W Felten. No silver bullet: De-identification still doesn't work. Jul 2014.

[112] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008.

[113] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. *2009 30th IEEE Symposium on Security and Privacy*, 2009.

[114] Lily Hay Newman. Ransomware hits dozens of hospitals in an unprecedented wave. *Wired*, Oct 2020.

[115] UK NHS. What the app does. *NHS COVID-19 app support*, 2020. Available at https://covid19.nhs.uk/what-the-app-does.html.

[116] Salvador Ochoa, Jamie Rasmussen, Christine Robson, and Michael Salib. Reidentification of individuals in chicago's homicide database: A technical and legal study. *Massachusetts Institute of Technology*, 08 2002.

[117] Government of Bahrain. Beaware bahrain. *Kingdom of Bahrain - eGovernment Apps Store*, 2020. Available at https://apps.bahrain.bh/CMSWebApplication/action/ShowAppDetailsAction?selectedAppID=321&appLanguage=en⟩.

[118] Government of Denmark. Smitte:stop answer questions about the app. *smittestop.dk*, 2020. Available at https://smittestop.dk/spoergsmaal-og-svar/.

[119] Ministry of Electronics Information Technology. Aarogyasetu bug bounty programme (for android app). *Bug Bounty Program*, May 2020.

[120] Finland Department of Health and Welfare. koronavilkku frequently asked. *koronavilkku.fi*, 2020. Available at https://koronavilkku.fi/ukk/.

[121] Israel Ministry of Health. Privacy policy and information security. *gov.il*, 2020. Available at https://govextra.gov.il/ministry-of-health/hamagen-app/magen-privacy-en/.

[122] New Zealand Ministry of Health. Privacy and security statement. *tracing.covid19.govt.nz*, 2020. Available at https://tracing.covid19.govt.nz/help/privacypolicy.

[123] The Directorate of Health, The Department of Civil Protection, and Emergency Management. privacy statement: Covid-19 tracing app. *Upplýsingar um Covid-19 á Íslandi*.

[124] Ministry of Health of the Czech Republic. Terms and conditions – erouška. *erouska.cz*, 2020. Available at https://erouska.cz/en/podminky-pouzivani.

[125] National Informatics Center of India. Aarogya setu faq's. *Aarogya Setu*, 2020. Available at https://aarogyasetu.gov.in/faq/.

[126] Government of Russia. Contact tracer is a covid-19 risk tracking application. *contacttracer.ru*, 2020. Available at https://contacttracer.ru/app#rec178736030.

[127] Government of Singapore. Blue trace protocol. *bluetrace.io*, 2020. Available at https://bluetrace.io/.

[128] Office of the Privacy Commissioner of Canada. A framework for the government of canada to assess privacy-impactful initiatives in response to covid-19. *Office of the Privacy Commissioner of Canada*, Apr 2020.

[129] Office of the Privacy Commissioner of Canada. Privacy review of the covid alert exposure notification application. *Office of the Privacy Commissioner of Canada*, Jul 2020.

[130] City of Toronto. Partner notification responsibility for health professionals. *toronto.ca*, Mar 2020. Available at https://www.toronto.ca/community-people/health-wellness-care/information-for-healthcare-professionals/sexual-health-info-for-health-professionals/partner-notification/.

[131] World Health Organization. Timeline: Who's covid-19 response. *who.int*, Jul 2020.

[132] Sharon Otterman. N.y.c. hired 3,000 workers for contact tracing. it's off to a slow start. *The New York Times*, Jun 2020.

[133] Vijay Pandurangan. On taxis and rainbows. *Medium*, Jun 2014. Available at https://tech.vijayp.ca/of-taxis-and-rainbows-f6bc289679a1.

[134] The Canadian Internet Policy and Public Interest Clinic. On the data trail: How detailed information about you gets into the hands of organizations with whom you have no relationship. Apr 2006.

[135] V. Primault, S. Ben Mokhtar, C. Lauradoux, and L. Brunie. Time distortion anonymization for the publication of mobility data with high utility. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 539–546, Aug 2015.

[136] Mishaal Rahman. Here are the countries using google and apple's covid-19 contact tracing api. *xda*, Aug 2020.

[137] Eric Rescorla. Looking at designs for covid-19 contact tracing apps. *The Mozilla Blog*, Apr 2020.

[138] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications*, 10(1), 2019.

[139] Shrivastava Saurabh and Shrivastava Prateek. Role of contact tracing in containing the 2014 ebola outbreak: a review. *African health sciences*, Mar 2017.

[140] B. Schneier. Schneier on security. *Blog*, Dec 1999.

[141] NHS Scotland. Privacy notice for the protect scotland app. *protect.scot*, 2020. Available at https://protect.scot/privacy-policy-app.

[142] Ananda G Shankar, Kulsum Janmohamed, Babatunde Olowokure, Gillian E Smith, Angela H Hogan, Valerie De Souza, Anders Wallensten, Isabel Oliver, Oliver Blatchford, Paul Cleary, and et al. Contact tracing for influenza a(h1n1)pdm09 virus-infected passenger on international flight. *National Library of Medicine*.

[143] Ian Sherr. Apple listens to some siri recordings to make it better. *CNET*, Jul 2019.

[144] Tom Simonite. Who's listening when you talk to your google assistant? *Wired*, Jul 2019.

[145] Liz Sly. U.s. soldiers are revealing sensitive and dangerous information by jogging. *The Washington Post*, Jan 2018.

[146] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[147] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1261–1269, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[148] L Sweeney, M Von Loewenfeldt, and M Perry. Saying it's anonymous doesn't make it so: Re-identifications of "anonymized" law school data. *Technology Science*, 11 2018.

[149] L. Sweeney, J. S. Yoo, L. Perovich, K. E. Boronow, P. Brown, and J. G. Brody. Re-identification risks in hipaa safe harbor data: A study of data from one environmental health study. *Technology science*, 2017.

[150] Latanya Sweeney. Simple demographics often identify people uniquely. *Carnegia Mellon University*, 2000.

[151] LATANYA SWEENEY. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[152] Latanya Sweeney. Patient Identifiability in Pharmaceutical Marketing Data. *Carnegie Mellon University*, Jan 2011.

[153] Latanya Sweeney. Matching known patients to health records in washington state data. *SSRN Electronic Journal*, 2013.

[154] Latanya Sweeney, Akua Abu, and Julia Winn. Identifying participants in the personal genome project by name. *SSRN Electronic Journal*, 2013.

[155] Nazanin Takbiri, Amir Houmansadr, Dennis L. Goeckel, and Hossein Pishro-Nik. Limits of location privacy under anonymization and obfuscation. *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017.

[156] Nazanin Takbiri, Amir Houmansadr, Dennis L. Goeckel, and Hossein Pishro-Nik. Privacy against statistical matching: Inter-user correlation. *CoRR*, abs/1805.01296, 2018.

[157] Vanessa Teague, Chris Culnane, Eleanor McMurtry, and Robert Merkel. Tracing the challenges of covidsafe. *GitHub*, Apr 2020.

[158] PRIVATICS team Inria and Fraunhofer AISEC. Robert-proximity-tracing/documents. *ROBERT: ROBust and privacy-presERving proximity Tracing*, May 2020.

[159] Nora Von Thenen, Erman Ayday, and A Ercument Cicek. Re-identification of individuals in genomic data-sharing beacons via allele inference. *Bioinformatics*, 35(3):365–371, 2018.

[160] Brian Thompson and Danfeng Yao. The union-split algorithm and cluster-based anonymization of social networks. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 218–227, New York, NY, USA, 2009. ACM.

[161] Anthony Tockar. Riding with the stars: Passenger privacy in the nyc taxicab dataset. *Riding with the Stars: Passenger Privacy in the NYC Taxicab Dataset –*, Sep 2014.

[162] Carmela Troncoso, Mathias Payer, Jean-Pierre Hubaux, Marcel Salathé, James Larus, Edouard Bugnion, Wouter Lueks, Theresa Stadler, Apostolos Pyrgelis, Daniele Antonioli, Ludovic Barman, Sylvain Chatel, Kenneth Paterson, Srdjan Čapkun, David Basin, Jan Beutel, Dennis Jackson, Marc Roeschlin, Patrick Leu, Bart Preneel, Nigel Smart, Aysajan Abidin, Seda Gürses, Michael Veale, Cas Cremers, Michael Backes, Nils Ole Tippenhauer, Reuben Binns, Ciro Cattuto, Alain Barrat, Dario Fiore, Manuel Barbosa, Rui Oliveira, and José Pereira. Decentralized privacy-preserving proximity tracing. May 2020.

[163] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. In *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pages 94–94, April 2006.

[164] David T.S. Personal information protection and electronic documents act. 2000.

[165] Human Rights Watch. Russia: Intrusive tracking app wrongly fines muscovites. *Human Rights Watch*, Oct 2020.

[166] WHO. Contact tracing. *World Health Organization NewsRoom QA*, May 2017.

[167] WHO. Transmission of sars-cov-2: implications for infection prevention precautions. *World Health Organization*, Jul 2020.

[168] Wikipedia. Covid-19 apps. *Wikipedia*, Nov 2020.

[169] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *proceedings of the 2008 SIAM International Conference on Data Mining*, pages 739–750. SIAM, 2008.

[170] Hui Zang and Jean Bolot. Anonymization of location data does not work. *Proceedings of the 17th annual international conference on Mobile computing and networking - MobiCom 11*, 2011.

[171] Mark Zastrow. South korea is reporting intimate details of covid-19 cases: has it helped? *Nature News*, Mar 2020.

Appendix A

Resources For Assessing Contact Tracing Applications

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
Australia COVIDSafe	Government Website: https://www.health.gov.au/resources/apps-and-tools/covidsafe-app Background on CovidSafe: https://covidsafe.gov.au/background.html#privacy-security App source code: https://github.com/AU-COVIDSafe Privacy Policy: https://covidsafe.gov.au/privacy-policy.html
Austria Stopp Corona	source code: https://github.com/austrianredcross FAQ: https://www.stopp-corona.at/faq_stopp_corona_app/
Azerbaijan e-Tabib	Government Website: https://koronavirusinfo.az/az/page/haqqimizda/e-tebib-mobil-tetbiqini-yukleyin

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
Bahrain BeAware Bahrain	Download Page: https://apps.bahrain.bh/CMSWebApplication/action/ShowAppDetailsAction?selectedAppID=321&appLanguage=en BBC article: https://www.bbc.com/news/av/world-middle-east-53058669
Bangladesh Corona Tracer BD	Article: https://tbsnews.net/tech/covid-tracer-app-tests-negative-125629 Article: https://www.dhakatribune.com/bangladesh/2020/06/06/sohoz-develops-corona-tracer-bd-app-to-ensure-people-s-safety Article: https://www.newagebd.net/article/107813/ict-division-launches-corona-tracer-bd
Canada Covid Alert	OPC review: https://www.priv.gc.ca/en/privacy-topics/health-genetic-and-other-body-information/health-emergencies/rev_covid-app/ Blackberry review: https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert/help.html#9 Privacy Assessment: https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert/privacy-policy/assessment.html Open source codebase: https://github.com/CovidShield Privacy Policy: https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert/privacy-policy.html Group testing app: https://resources.securitycompass.com/blog/canada-covid-alert-app-android-security
China Health Code	New York Times: https://www.nytimes.com/2020/03/01/business/china-coronavirus-surveillance.html

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Business Insider: https://www.businessinsider.com/coronavirus-china-health-software-color-coded-how-it-works-2020-4 Wired Article: https://www.wired.co.uk/article/china-coronavirus-health-code-qr Review of documents: https://technode.com/2020/07/10/we-read-the-technical-standards-for-chinas-health-code-heres-what-we-learned/ Review of app: https://chinaindianetworked.substack.com/p/cin-14-going-under-the-hood-of-health BBC article: https://www.bbc.com/news/world-asia-india-52659520
Colombia CoronApp	Government Website: https://coronaviruscolombia.gov.co/Covid19/aislamiento-saludable/coronapp.html CDC: https://www.cdc.gov/globalhealth/healthprotection/fetp-40th-anniversary/stories/colombia-app-covid.html Google app store: https://play.google.com/store/apps/details?id=co.gov.ins.guardianes&hl=en Bogota Post: https://thebogotapost.com/tracking-coronavirus-coronapp/46864/ Privacy International: https://privacyinternational.org/examples/3435/colombia-coronapp-fails-public-information-purpose ITS Rio: https://itsrio.org/wp-content/uploads/2020/05/Coronapp-1.pdf
Croatia Stop Covid-19	Croatia News: https://www.total-croatia-news.com/news/45331-croatia-presents-its-stop-covid-19-app
Czech Republic eRouška	Google app store: https://play.google.com/store/apps/details?id=cz.covid19cz.erouska&hl=en_US Government Website: https://erouska.cz/

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	**Resources Used**
	Audits: https://erouska.cz/audit-kod Source Code: https://github.com/covid19cz Commitment to data trust: https://www.covid19cz.cz/covid19-cz/zavazek-datove-duvery Privacy Statement: https://erouska.cz/en/gdpr
Denmark Smittestop	Government Website: https://smittestop.dk/ FAQ: https://smittestop.dk/spoergsmaal-og-svar/ Impact Assessment: https://smittestop.dk/uploads/konsekvensanalyse_vedr_databeskyttelse.pdf Personal Data Statement: https://smittestop.dk/databeskyttelse/
Ecuador SO Covid Tool	Government Wesbite: https://asiecuador.com/ Human Rights Watch: https://www.hrw.org/news/2020/07/01/ecuador-privacy-risk-covid-19-surveillance National Post: https://nationalpost.com/pmn/health-pmn/ ecuador-mobilizes-covid-19-watchers-to-contain-pandemic-in-the-capital iOS app store: https://apps.apple.com/app/id1523594087 Google app store: https://play.google.com/store/apps/details?id=ec.gob.asi.android Primicias https://www.primicias.ec/noticias/tecnologia/asi-ecuador-aplicacion-rastrear-covid/ El Universo: https://www.eluniverso.com/noticias/2020/08/03/nota/7929402/ asi-ecuador-app-monitoreo-coronavirus-bluetooth Open source code: https://minka.gob.ec/asi-ecuador
Estonia Hoia	Google App store: https://play.google.com/store/apps/details?id=ee.tehik.hoia iOS App store: https://apps.apple.com/app/id1515441601 Government Website: https://hoia.me/en/

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Privacy Policy: ⟨https://hoia.me/privacy/ Source Code: https://koodivaramu.eesti.ee/tehik/hoia
Ethiopia Debo	Google app store: https://play.google.com/store/apps/details?id=com.ewenet.debo&hl=en_CA Government Website: https://debo.ephi.gov.et/ VOA News: https://www.voanews.com/covid-19-pandemic/ ethiopian-diaspora-champions-digital-apps-fight-against-covid Covid response: https://www.jsi.com/ethiopias-digital-health-response-to-covid-19/
France TousAntiCovid	iOS app store: https://apps.apple.com/app/id1511279125 Google app store: https://play.google.com/store/apps/details?id=fr.gouv.android.stopcovid Government Website: https://www.gouvernement.fr/info-coronavirus/tousanticovid Government Info: https://solidarites-sante.gouv.fr/soins-et-maladies/maladies/ maladies-infectieuses/coronavirus/tousanticovid Source code: https://gitlab.inria.fr/stopcovid19
Fiji careFIJI	iOS app store: https://apps.apple.com/fj/app/carefiji/id1513752467#?platform=ipad Google app store: https://play.google.com/store/apps/details?id=fj.gov.carefiji&hl=en Government Website: https://carefiji.digitalfiji.gov.fj/ FAQ: https://carefiji.digitalfiji.gov.fj/faqs/ Fact sheet: https://carefiji.digitalfiji.gov.fj/wp-content/uploads/2020/09/Fact-Sheet-careFIJI-20200831.pdf Privacy Policy: https://carefiji.digitalfiji.gov.fj/privacy-policy/
Finland Koronavilkku	Google App store: https://play.google.com/store/apps/details?id=fi.thl.koronahaavi

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	**Resources Used**
	iOS app store: https://apps.apple.com/fi/app/id1520576224 Government Website: https://koronavilkku.fi/ FAQ: https://koronavilkku.fi/ukk/ Source code: https://github.com/THLfi
Germany Corona-Warn App	Google App store: https://play.google.com/store/apps/details?id=de.rki.coronawarnapp iOS app store: https://apps.apple.com/de/app/corona-warn-app/id1512595757 Government Website: https://www.coronawarn.app/en/ FAQ: https://www.coronawarn.app/en/faq/ Source code: https://github.com/corona-warn-app Privacy Notice: https://www.coronawarn.app/assets/documents/cwa-privacy-notice-en.pdf Privacy Statement: https://www.coronawarn.app/en/privacy/
Ghana GH Covid-19 Tracker App	Google App Store: https://play.google.com/store/apps/details?id=com.moc.gh&hl=en_CA iOS app store: https://apps.apple.com/gh/app/gh-covid-19-tracker/id1508568320 Government Website: https://ghcovid19.com/ CGTN News: https://africa.cgtn.com/2020/04/24/ghana-develops-app-to-help-track-covid-19-patients/
Gibraltar Beat Covid Gibraltar	Google App store: https://play.google.com/store/apps/details?id=com.gha.covid.tracker iOS app store: https://apps.apple.com/gb/app/beat-covid-gibraltar/id1514587092 Government Website: https://www.gibraltar.gov.gi/beatcovidapp
Guatemala Alerta Guate	Global Witness: https://www.globalwitness.org/en/campaigns/covid-19-tracing-apps-must-not-interfere-human-rights/

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Company Website: https://in-telligent.com/apps/alerta-guate/ Company Privacy Policy: https://in-telligent.com/application-privacy-policy-2/
Hungary VirusRadar	Google app store: https://play.google.com/store/apps/details?id=hu.gov.virusradar Government Website: https://virusradar.hu/ Developer Website: https://www.nextsense.com/ns-newsarticle-virusradar-a-mobile-contact-tracing-implemented.nspx
Iceland Rakning C-19	Google App Store: https://play.google.com/store/apps/details?id=is.landlaeknir.rakning&hl=en_CA iOS: https://apps.apple.com/gh/app/rakning-c-19/id1504655876 Government Website: https://www.covid.is/app/en Privacy Statement: https://www.covid.is/app/privacystatement Source code: https://github.com/aranja/rakning-c19-app
India Aarogya Setu	Google App store: https://play.google.com/store/apps/details?id=nic.goi.aarogyasetu&hl=en_CA iOS app store: https://apps.apple.com/in/app/aarogyasetu/id1505825357 Government Website: https://aarogyasetu.gov.in/ Bug Bounty Program: https://static.mygov.in/rest/s3fs-public/mygov_159057669351307401.pdf FAQ: https://aarogyasetu.gov.in/faq/ Technical FAQ: https://aarogyasetu.gov.in/technical-faqs/ Privacy Policy: https://aarogyasetu.gov.in/privacy-policy/ privacy Statement: https://aarogyasetu.gov.in/wp-content/uploads/2020/06/mygov-1000000000981057882.pdf App source code: https://github.com/nic-delhi

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	BBC article: https://www.bbc.com/news/world-asia-india-52659520 Hacking Documentation: https://medium.com/@fs0c131y/aarogya-setu-the-story-of-a-failure-3a190a18e34 Hacking Documentation: https://www.buzzfeednews.com/article/pranavdixit/india-aarogya-setu-hacked Data access statement: https://aarogyasetu.gov.in/wp-content/uploads/2020/06/mygov-1000000000981057882.pdf Updates to functions: https://www.medianama.com/2020/07/223-aarogya-setu-bluetooth-contacts/
Ireland COVID Tracker	Google App Store: https://play.google.com/store/apps/details?id=com.covidtracker.hse iOS app store: https://apps.apple.com/ie/app/covid-tracker-ireland/id1505596721 Source code: https://github.com/HSEIreland/ Government Website: https://covidtracker.gov.ie/ Privacy Notice: https://covidtracker.gov.ie/privacy-and-data/data-protection/#11
Israel HaMagen	Google App Store: https://play.google.com/store/apps/details?id=com.hamagen iOS App Store: https://apps.apple.com/us/app/id1503224314?ls=1 Government Website: https://govextra.gov.il/ministry-of-health/hamagen-app/download-en/ FAQ: https://govextra.gov.il/ministry-of-health/hamagen-app/magen-faq-en/ Privacy Policy: https://govextra.gov.il/ministry-of-health/hamagen-app/privacy-policy-en/ Privacy and Secuirty Statement: https://govextra.gov.il/ministry-of-health/hamagen-app/magen-privacy-en/ Source Code: https://github.com/MohGovIL/hamagen-react-native

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
Italy Immuni	Google App Store: https://play.google.com/store/apps/details?id=it.ministerodellasalute.immuni iOS App Store: https://apps.apple.com/app/id1513940977 Government Website: https://www.immuni.italia.it/ Source Code: https://github.com/immuni-app FAQ: https://www.immuni.italia.it/faq.html
Japan COVID-19 Contact-Confirming Application	Google app Store: https://play.google.com/store/apps/details?id=jp.go.mhlw.covid19radar iOS app store: https://apps.apple.com/jp/app/covid-19-contact-app/id1516764458?l=en Source code: https://github.com/Covid-19Radar/Covid19Radar FAQ: https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryou/covid19_qa_kanrenkigyou_00009.html Privacy Policy: https://www.mhlw.go.jp/stf/seisakunitsuite/english_pp_00032.html
Jordan AMAN App - Jordan	Google app store: https://play.google.com/store/apps/details?id=jo.gov.moh.aman&hl=en_CA iOS app store: https://apps.apple.com/us/app/aman-aman-jo-jordan-covid-19/id1511595289 Huawei app store: https://appgallery.huawei.com/#/app/C102442027?locale=en_US&source=appshare&subsource=C102442027 Government Website: https://amanapp.jo/en FAQ: https://amanapp.jo/en/page/12/FAQs Privacy Policy: https://amanapp.jo/en/page/10/Privacy About the app: https://amanapp.jo/en/page/8/Privacy#mainTitle
Kazakhstan eGov bizbirgemiz mobile app	iOS: https://apps.apple.com/ai/app/egov-mobile/id1476128386 User Agreement: https://egov.kz/cms/en/articles/aggreement_mobile

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Privacy Policy: https://egov.kz/cms/en/articles/privacy_policy_eGov_bizbirgemiz
Kuwait Shlonik	Google App store: https://play.google.com/store/apps/details?id=com.healthcarekw.app&hl=en iOS app store: https://apps.apple.com/za/app/shlonik-%D8%B4%D9%84%D9%88%D9%86%D9%83/id1503978984 Video of Registration: https://www.youtube.com/watch?v=26aX1bArMow BBC article: https://www.bbc.com/news/world-middle-east-53052395
Latvia Apturi Covid	Google App Store: https://play.google.com/store/apps/details?id=lv.spkc.gov.apturicovid iOS App Store: https://apps.apple.com/app/id1513573144 Government Website: https://www.apturicovid.lv/ FAQ: https://apturicovid.lv/biezak-uzdotie-jautajumi/#en Source code: https://github.com/ApturiCOVID
Malaysia MyTrace	Google App Store: https://play.google.com/store/apps/details?id=my.gov.onegovappstore.mytrace&hl=en Government Website: https://www.mosti.gov.my/web/en/mytrace/ Covid Response: https://themalaysianreserve.com/2020/05/12/three-major-apps-to-trace-covid-19/
Netherlands CoronaMelder	Google App Store: https://play.google.com/store/apps/details?id=nl.rijksoverheid.en iOS app store: https://apps.apple.com/nl/app/coronamelder/id1517652429 Government Website: https://coronamelder.nl/ FAQ: https://coronamelder.nl/nl/faq Privacy Statement: https://coronamelder.nl/nl/privacy

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Source Code: https://github.com/minvws
New Zealand NZ COVID Tracer	Google App Store: https://play.google.com/store/apps/details?id=nz.govt.health.covidtracer&hl=en_CA iOS App Store: https://apps.apple.com/nz/app/nz-covid-tracer/id1511667597 Government Website: https://tracing.covid19.govt.nz/ Privacy Statement: https://tracing.covid19.govt.nz/ Impact Assessment: https://www.health.govt.nz/our-work/diseases-and-conditions/covid-19-novel-coronavirus/covid-19-resources-and-tools/nz-covid-tracer-app/privacy-and-security-nz-covid-tracer
North Macedonia Stop Korona!	Google App Store: https://play.google.com/store/apps/details?id=mk.gov.koronavirus.stop&hl=en_CA iOS App Store: https://apps.apple.com/mk/app/stopkorona!/id1506641869 Government Website: https://stop.koronavirus.gov.mk/ Privacy Policy: https://stop.koronavirus.gov.mk/privacy-policy
Northern Ireland StopCOVID NI	Google App Store: https://play.google.com/store/apps/details?id=net.hscni.covidtracker iOS App Store: https://apps.apple.com/gb/app/stopcovidni/id1519404160 Government Website: https://covid-19.hscni.net/stop-covid-ni-mobile-app/ Source Code: https://github.com/HSCNI-GITHUB Privacy Information: https://covid-19.hscni.net/stopcovid-ni-privacy-information/ FAQ: https://covid-19.hscni.net/stopcovid-ni-faqs/

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
Norway Smittestopp	Amnesty International: https://www.amnesty.org/en/latest/news/2020/06/bahrain-kuwait-norway-contact-tracing-apps-danger-for-privacy/ TechCrunch article: https://techcrunch.com/2020/06/15/norway-pulls-its-coronavirus-contacts-tracing-app-after-privacy-watchdogs-warning/ Helsenorge Article: https://helsenorge.no/coronavirus/smittestopp Life in Norway: https://www.lifeinnorway.net/smittestopp-coronavirus-app/ Hacking Documentation: https://nrkbeta.no/2020/04/20/sa-enkelt-er-det-a-forfalske-sms-er-fra-smittestopp/
Poland ProteGO Safe	Google App Store: https://play.google.com/store/apps/details?id=pl.gov.mc.protegosafe iOS App Store: https://apps.apple.com/us/app/protego-safe/id1508481566 Government Website: https://www.gov.pl/web/protegosafe Source Code: https://github.com/ProteGO-Safe Documentation: https://github.com/ProteGO-Safe/specs/blob/master/README-ENG.md FAQ: https://www.gov.pl/web/protegosafe/pytania-i-odpowiedzi
Portugal STAYAWAY COVID	Google App Store: https://play.google.com/store/apps/details?id=fct.inesctec.stayaway iOS App Store: https://apps.apple.com/pt/app/id1519479652 Government Wesbite: https://stayawaycovid.pt/ Source Code: https://github.com/stayawayinesctec FAQ: https://stayawaycovid.pt/perguntas-frequentes/ Documentation: https://stayawaycovid.pt/wp-content/uploads/STAWAWAY-COVID-doc.pdf
Qatar Ehteraz app	Google App Store: https://play.google.com/store/apps/details?id=com.moi.covid19&hl=en

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	iOS App Store: https://apps.apple.com/us/app/ehteraz/id1507150431 Amnesty International: https://www.amnesty.org/en/latest/news/2020/06/bahrain-kuwait-norway-contact-tracing-apps-danger-for-privacy/ Vulnerability: https://www.amnesty.org/en/latest/news/2020/05/qatar-covid19-contact-tracing-app-security-flaw/ Aljazeera: https://www.aljazeera.com/news/2020/5/26/qatar-makes-covid-19-app-mandatory-experts-question-efficiency WIONews: https://www.wionews.com/world/qatar-coronavirus-contact-tracing-app-ehteraz-stirs-rare-privacy-backlash-300888
Russia Social Monitoring service	Human Rights Watch: https://www.hrw.org/news/2020/05/21/russia-intrusive-tracking-app-wrongly-fines-muscovites Moscow IT: https://www.mos.ru/dit/documents/view/238534220/ Moscow Mayor: https://www.interfax.ru/moscow/705313
Russia Contact Tracer	Government Website: https://contacttracer.ru/app#rec178736030
Saudia Arabia Tabaud	Google App Store: https://play.google.com/store/apps/details?id=sa.gov.nic.tabaud iOS App Store: https://apps.apple.com/sa/app/tabaud-covid-19-ksa/id1514704802 Goverment Website: https://tabaud.sdaia.gov.sa/IndexE FAQ: https://tabaud.sdaia.gov.sa/FAQEn Privacy Statement: https://tabaud.sdaia.gov.sa/PrivacyEn
Scotland Protect Scotland	Google App Store: https://play.google.com/store/apps/details?id=gov.scot.covidtracker iOS App Store: https://apps.apple.com/gb/app/id1526637715

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	**Resources Used**
	Government Website: https://protect.scot/ How it Works: https://protect.scot/how-it-works Privacy Notice: https://protect.scot/privacy-policy-app Data Use: https://protect.scot/how-we-use-your-data Terms and Conditions: https://protect.scot/terms-and-conditions
Singapore TraceTogether	Google App Store: https://play.google.com/store/apps/details?id=sg.gov.tech.bluetrace iOS App Store: https://apps.apple.com/sg/app/tracetogether/id1498276074 Government Website: https://www.tracetogether.gov.sg/ FAQ: https://support.tracetogether.gov.sg/hc/en-sg Privacy Statement: https://www.tracetogether.gov.sg/common/privacystatement Protocol: https://bluetrace.io/ Code Base: https://github.com/OpenTrace-community BBC Article: https://www.bbc.com/news/technology-53146360 User Walkthrough: https://www.moh.gov.sg/docs/librariesprovider5/tracetogether/how_to_upload_your_tracetogether_data_(english).pdf
Slovenia #OstaniZdrav	Google App Store: https://play.google.com/store/apps/details?id=si.gov.ostanizdrav Government Website: https://www.gov.si/en/topics/coronavirus-disease-covid-19/the-ostanizdrav-mobile-application/ Privacy Notice: https://www.gov.si/assets/vlada/Koronavirus-zbirno-infografike-vlada/APP-OstaniZdrav/Privacy-notice.pdf
South Africa COVID Alert South Africa	Google App Store: https://play.google.com/store/apps/details?id=za.gov.health.covidconnect iOS App Store: https://apps.apple.com/app/apple-store/id1524618326?mt=8

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	Government Website: https://sacoronavirus.co.za/ Information Page: https://sacoronavirus.co.za/covidalert/ Privacy Policy: https://sacoronavirus.co.za/covidalert/privacy-policy/
Spain Radar COVID	Google App Store: https://play.google.com/store/apps/details?id=es.gob.radarcovid iOS App Store: https://apps.apple.com/es/app/radar-covid/id1520443509 Government Website: https://twitter.com/AppRadarCovid Source Code: https://github.com/radarcovid FAQ: https://radarcovid.gob.es/preguntas-frecuentes Privacy Policy: https://radarcovid.gob.es/politica-de-privacidad Manifesto: https://radarcovid.gob.es/manifiesto
Switzerland SwissCovid app	Google App Store: SwitzerlandSwissCovidapp iOS App Store: https://apps.apple.com/us/app/swisscovid/id1509275381 App Source Code: https://github.com/DP-3T/dp3t-app-android-ch Data protection Statment: https://www.bag.admin.ch/bag/en/home/krankheiten/ausbrueche-epidemien-pandemien/ aktuelle-ausbrueche-epidemien/novel-cov/swisscovid-app-und-contact-tracing/ datenschutzerklaerung-nutzungsbedingungen.html#-11360452 Government Website: https://www.bag.admin.ch/bag/en/home/krankheiten/ ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov/ swisscovid-app-und-contact-tracing.html#-1032598416 Security Assessment: https://www.melani.admin.ch/SwissCovid_en
UK NHS COVID-19	Google App Store: https://play.google.com/store/apps/details?id=uk.nhs.covid19.production

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	**Resources Used**
	iOS: https://apps.apple.com/us/app/nhs-covid-19/id1520427663?mt=8&ign-mpt=uo%3D4 Source Code: https://github.com/nhsx Government Website: https://covid19.nhs.uk/ About Data and Privacy Statement: https://covid19.nhs.uk/privacy-and-data.html Privacy Notice: https://www.gov.uk/government/publications/nhs-covid-19-app-privacy-information What the App Does: https://covid19.nhs.uk/what-the-app-does.html FAQ: https://faq.covid19.nhs.uk/ Privacy Notice: https://www.gov.uk/government/publications/nhs-covid-19-app-privacy-information/ nhs-test-and-trace-app-early-adopter-trial-august-2020-privacy-notice
Turkey Life Fits Inside the Home	Google App Store: https://play.google.com/store/apps/details?id=tr.gov.saglik.hayatevesigar&hl=en_US&gl=US iOS: https://apps.apple.com/tr/app/hayat-eve-s%C4%B1%C4%9Far/id1505756398?l=tr Government: https://hayatevesigar.saglik.gov.tr/ Help Page: https://hayatevesigar.saglik.gov.tr/hes-eng.html Turkish Airlines Statement: https://www.turkishairlines.com/en-int/announcements/coronavirus-outbreak/hes-code/
USA CoEpi	Company Website: https://www.coepi.org/ Source Code: https://github.com/Co-Epi
USA CovidSafe/CommonCircle	Washington University: https://covidsafe.cs.washington.edu/ Project Website: https://commoncircle.us/

Table A.1: Resources for used while assessing contact tracing apps

Country and Application	Resources Used
	White Paper: https://commoncircle.us/Whitepaper.html Source Code: https://github.com/CovidSafe
USA Covid Watch	Google App Store: https://play.google.com/store/apps/details?id=gov.azdhs.covidwatch.android iOS App Store: https://apps.apple.com/us/app/id1521655110 Source Code: https://github.com/covidwatchorg Company Website: https://www.covidwatch.org/ FAQ: https://www.covidwatch.org/faq
USA California COVID Notify	Google App Store: https://play.google.com/store/apps/details?id=gov.ca.covid19.exposurenotifications Government Website: https://covid19.ca.gov/notify/ Privacy Policy: https://covid19.ca.gov/notify-privacy/
USA Care19 Alert	Google App Store: https://play.google.com/store/apps/details?id=com.proudcrowd.exposure iOS App Store: https://apps.apple.com/us/app/care19-alert/id1513945072 Company Website: https://www.care19.app/ Privacy Statement: https://www.care19.app/Privacy

Curriculum Vitae

Name:	Leah Krehling
Post-Secondary Education and Degrees:	University of Windsor Windsor, ON 2014 - 2018 B.A.Sc University of Western Ontario London, ON 2019 - 2020 M.E.Sc.
Related Work Experience:	Teaching Assistant The University of Western Ontario 2019 - 2020

Publications:

L. Krehling, "De-Identification Guideline." Technical Report. Western Information Security and Privacy Research Laboratory, Western University, Canada, 2020. Available online: https://whisperlab.org/technical-reports/de-identification-guideline-WL-2020-01.pdf